PRANAV RAIKAR

D15A 47

Experiment 2

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');

}

}

The above code is used in following below code of home.dart

import 'package:flutter/material.dart';

import 'package:food\_panda/pages/details.dart';

import 'package:food\_panda/widget/widget\_support.dart';

class Home extends StatefulWidget {

const Home({super.key});

@override

State<Home> createState() => \_HomeState();

}

class \_HomeState extends State<Home> {

bool icecream = false, pizza = false, salad = false, burger = false;

@override

Widget build(BuildContext context) {

return Scaffold(

body: SingleChildScrollView(

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)),

child: const Icon(

Icons.shopping\_cart\_outlined,

color: Colors.white,

),

)

],

),

const SizedBox(

height: 20.0,

),

Text("Delicious Food", style: AppWidget.HeadlineTextFieldStyle()),

Text("Discover and Get Great Food",

style: AppWidget.LightTextFieldStyle()),

const SizedBox(

height: 20.0,

),

Container(

margin: const EdgeInsets.only(right: 20.0),

child: showItem()),

const SizedBox(

height: 30.0,

),

SingleChildScrollView(

scrollDirection: Axis.horizontal,

child: Row(

children: [

GestureDetector(

onTap: () {

Navigator.push(context,

MaterialPageRoute(builder: (context) => Details()));

},

child: Container(

margin: const EdgeInsets.all(4),

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(20),

child: Container(

padding: const EdgeInsets.all(14),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Image.asset(

"images/salad2.png",

height: 150,

width: 150,

fit: BoxFit.cover,

),

Text("Veggie Taco Hash",

style:

AppWidget.semiBoldTextFieldStyle()),

const SizedBox(

height: 5.0,

),

Text("Fresh and Healthy",

style: AppWidget.LightTextFieldStyle()),

const SizedBox(

height: 5.0,

),

Text(

"\$25",

style: AppWidget.semiBoldTextFieldStyle(),

)

]),

),

),

),

),

const SizedBox(

width: 15.0,

),

Container(

margin: const EdgeInsets.all(4),

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(20),

child: Container(

padding: const EdgeInsets.all(14),

child: Column(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Image.asset(

"images/salad4.png",

height: 150,

width: 150,

fit: BoxFit.cover,

),

Text("Mix Veg Salad",

style: AppWidget.semiBoldTextFieldStyle()),

const SizedBox(

height: 5.0,

),

Text("Spicy with Onion",

style: AppWidget.LightTextFieldStyle()),

const SizedBox(

height: 5.0,

),

Text(

"\$28",

style: AppWidget.semiBoldTextFieldStyle(),

)

]),

),

),

),

],

),

),

const SizedBox(

height: 30.0,

),

Container(

margin: const EdgeInsets.only(right: 20.0),

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(20),

child: Container(

padding: const EdgeInsets.all(5),

child: Row(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Image.asset(

"images/salad4.png",

height: 120,

width: 120,

fit: BoxFit.cover,

),

const SizedBox(

width: 20.0,

),

Column(

children: [

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"Mediterranean Chickpea Salad",

style: AppWidget.semiBoldTextFieldStyle(),

)),

const SizedBox(

height: 5.0,

),

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"Honey goot cheese",

style: AppWidget.LightTextFieldStyle(),

)),

const SizedBox(

height: 5.0,

),

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"\$28",

style: AppWidget.semiBoldTextFieldStyle(),

))

],

)

],

),

),

),

),

const SizedBox(

height: 30.0,

),

Container(

margin: const EdgeInsets.only(right: 20.0),

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(20),

child: Container(

padding: const EdgeInsets.all(5),

child: Row(

crossAxisAlignment: CrossAxisAlignment.start,

children: [

Image.asset(

"images/salad2.png",

height: 120,

width: 120,

fit: BoxFit.cover,

),

const SizedBox(

width: 20.0,

),

Column(

children: [

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"Veggie Taco Hash",

style: AppWidget.semiBoldTextFieldStyle(),

)),

const SizedBox(

height: 5.0,

),

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"Honey goot cheese",

style: AppWidget.LightTextFieldStyle(),

)),

const SizedBox(

height: 5.0,

),

Container(

width: MediaQuery.of(context).size.width / 2,

child: Text(

"\$28",

style: AppWidget.semiBoldTextFieldStyle(),

))

],

)

],

),

),

),

),

],

),

),

),

);

}

Widget showItem() {

return Row(

mainAxisAlignment: MainAxisAlignment.spaceBetween,

children: [

GestureDetector(

onTap: () {

icecream = true;

pizza = false;

salad = false;

burger = false;

setState(() {});

},

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(10),

child: Container(

decoration: BoxDecoration(

color: icecream ? Colors.black : Colors.white,

borderRadius: BorderRadius.circular(10)),

padding: const EdgeInsets.all(8),

child: Image.asset(

"images/ice-cream.png",

height: 40,

width: 40,

fit: BoxFit.cover,

color: icecream ? Colors.white : Colors.black,

),

),

),

),

GestureDetector(

onTap: () {

icecream = false;

pizza = true;

salad = false;

burger = false;

setState(() {});

},

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(10),

child: Container(

decoration: BoxDecoration(

color: pizza ? Colors.black : Colors.white,

borderRadius: BorderRadius.circular(10)),

padding: const EdgeInsets.all(8),

child: Image.asset(

"images/pizza.png",

height: 40,

width: 40,

fit: BoxFit.cover,

color: pizza ? Colors.white : Colors.black,

),

),

),

),

GestureDetector(

onTap: () {

icecream = false;

pizza = false;

salad = true;

burger = false;

setState(() {});

},

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(10),

child: Container(

decoration: BoxDecoration(

color: salad ? Colors.black : Colors.white,

borderRadius: BorderRadius.circular(10)),

padding: const EdgeInsets.all(8),

child: Image.asset(

"images/salad.png",

height: 40,

width: 40,

fit: BoxFit.cover,

color: salad ? Colors.white : Colors.black,

),

),

),

),

GestureDetector(

onTap: () {

icecream = false;

pizza = false;

salad = false;

burger = true;

setState(() {});

},

child: Material(

elevation: 5.0,

borderRadius: BorderRadius.circular(10),

child: Container(

decoration: BoxDecoration(

color: burger ? Colors.black : Colors.white,

borderRadius: BorderRadius.circular(10)),

padding: const EdgeInsets.all(8),

child: Image.asset(

"images/burger.png",

height: 40,

width: 40,

fit: BoxFit.cover,

color: burger ? Colors.white : Colors.black,

),

),

),

),

],

);

}

}

A screenshot of a phone

Description automatically generated

Conclusion :Thus I learnt to create and use common widgets