TIERS LIMITED SUMMER INTERNSHIP 2024

MOBILE APP DEVELOPMENT

Advance UI Components

(Continue)

CLASS # 13

TIERS
Limited

Overview

In this class, we will cover advanced Flutter widgets such as Stack, Radio, and Positioned. These widgets are essential for creating complex and interactive user interfaces in Flutter applications.

Objectives

- Understand the purpose and usage of the Stack widget.
- Explore the Positioned widget for precise control over widget positioning within a Stack.
- Learn how to use the Radio widget for selection input.

Stack Widget

Purpose

The Stack widget allows you to overlay multiple widgets on top of each other. It is useful for creating complex layouts where widgets need to overlap.

Usage

- The Stack widget arranges its children widgets in a back-to-front order.
- You can use Positioned widgets within a Stack to control the exact position of each child.

Example:

```
Stack(
      children: <Widget>[
        Container(
          width: 200,
          height: 200,
          color: Colors.red,
        ),
        Container(
          width: 150,
          height: 150,
          color: Colors.green,
        ),
        Container(
          width: 100,
          height: 100,
          color: Colors.blue,
        ),
      ],
    ),
  ),
);
```

Positioned Widget

Purpose

The Positioned widget is used within a Stack to position a child widget at a specific location. It provides precise control over where the child widget appears.

Usage

- The Positioned widget must be a direct child of a Stack.
- Use the top, bottom, left, and right properties to specify the position.

Example

```
children: <Widget>[
  Container(
    width: 200,
    height: 200,
    color: Colors.red,
  ),
  Positioned(
    top: 50,
    left: 50,
    child: Container(
      width: 100,
      height: 100,
      color: Colors.green,
    ),
  ),
],
```

Radio Widget

Purpose

The Radio widget allows users to select a single option from a set of options. It is commonly used in forms and surveys.

Usage

- Use the Radio widget along with a state variable to manage the selected option.
- Each Radio widget should have a unique value.

Example

```
int _selectedValue = 1;
```

```
RadioListTile<int>(
  title: Text('Option 1'),
  value: 1,
  groupValue: _selectedValue,
  onChanged: (value) {
    setState(() {
      _selectedValue = value!;
    });
  },
),
RadioListTile<int>(
  title: Text('Option 2'),
  value: 2,
  groupValue: _selectedValue,
  onChanged: (value) {
    setState(() {
      _selectedValue = value!;
    });
  },
),
```

Exercises:

Exercise 1: Creating a Stack Layout using Positioned Widget

Exercise 2: Using Radio Widgets

- Create a form with multiple Radio widgets allowing the user to select a single option from a set of choices.