LAB SESSION 12

Stack & Grid View Widget

Objective:

The objective of this lab is to provide hands-on experience and deepen understanding of the Stack and GridView widgets in Flutter.

Introduction:

The Stack widget in Flutter allows you to overlay multiple widgets on top of each other. This is useful for creating complex layouts where widgets can overlap or need to be positioned relative to each other. The Stack widget does not constrain its children, meaning the children can be positioned anywhere within the stack.

Key Concepts:

· Alignment: Controls how the children are aligned within the Stack.
· Positioned: A widget used within a Stack to position its child relative to the stack's edges. · Overflow: Controls how the stack should behave if children overflow its bounds (deprecated in favor of clipBehavior).

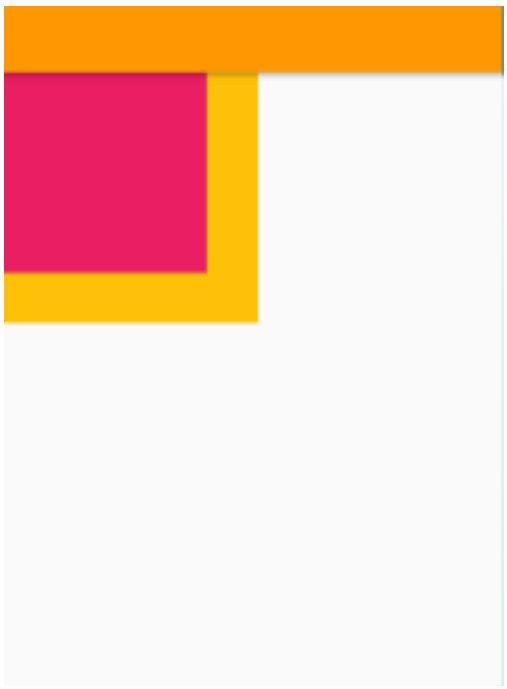
Properties:

· alignment: Aligns the non-positioned children within the stack. Default is AlignmentDirectional.topStart.
· fit: Determines how non-positioned children should be sized. Values can be StackFit.loose, StackFit.expand, or StackFit.passthrough.
· overflow: Previously controlled overflow behavior, now replaced by clipBehavior. · clipBehavior: Defines how to handle children that are outside the stack's bounds. Values include Clip.none, Clip.hardEdge, Clip.antiAlias, and Clip.antiAliasWithSaveLayer.

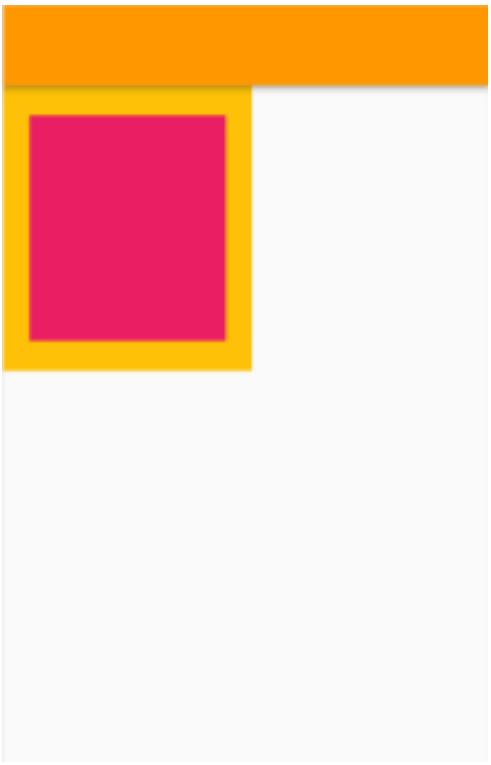
Usage:

· Non-positioned children are aligned according to the alignment property. · Positioned children are placed based on the properties like top, left, right, and bottom.

```
body:
    Stack(
        children: [
          Container(height:200,
          width:200,
          color:Colors.amber),
          Container(height:160,
          width:160,
          color:Colors.pink),


        ],

      )
```



```
body:
Stack(
 children: [
 Container(height:200,
  width:200,
  color:Colors.amber),
  Positioned(
   top:21,
   left:21,
   right:21,
   bottom:21,
   child:Container(height:160,
   width:160,
    color:Colors.pink),

)],)
```



Grid View:

The GridView class in Flutter is used to create scrollable grids of widgets. It is particularly useful for displaying a large number of items in a grid pattern, where each item can be customized. The GridView widget provides a variety of constructors to create different types of grids.

Key Concepts:

1. **GridDelegate**: A delegate that controls the layout of the children within the grid. Commonly used delegates include SliverGridDelegateWithFixedCrossAxisCount and SliverGridDelegateWithMaxCrossAxisExtent.
2. **ScrollDirection**: Defines the axis along which the grid scrolls (vertical or horizontal). 3. **Primary**: Determines if this is the primary scroll view associated with the parent primary scroll controller.

4. **Padding**: Adds padding around the grid content.
5. **Physics**: Controls how the scroll view behaves (e.g., bouncing, clamping). 6. **ShrinkWrap**: Controls whether the grid should take only as much space as needed or fill its parent.

Constructors:

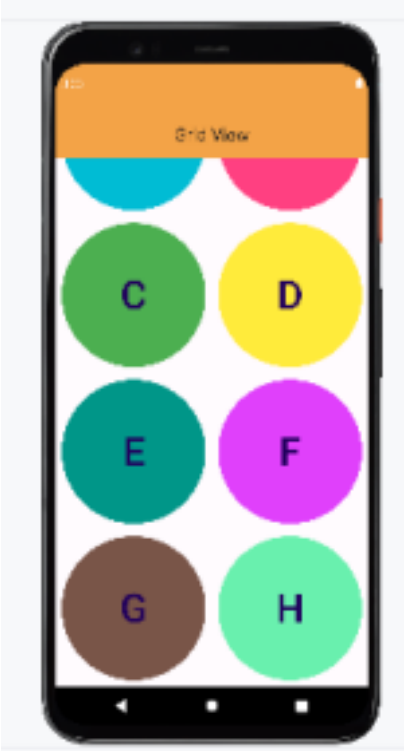
1. **GridView**: The default constructor for creating a grid view with a given SliverGridDelegate.
2. **GridView.count**: Creates a grid view with a fixed number of columns. 3. **GridView.extent**: Creates a grid view with tiles that have a maximum cross-axis extent. 4. **GridView.builder**: Creates a grid view using a builder pattern, which is efficient for large grids.
5. **GridView.custom**: Creates a grid view with a custom child layout and scroll behavior.


```
body:
 GridView.count(crossAxisCount: 3,

 children: [
 Container(

 child: CircleAvatar(backgroundColor: Colors.cyan,
 child:Text('A',style: TextStyle(fontSize: 50,),),),),
 Container(
```



```
 child: CircleAvatar(backgroundColor: Colors.pinkAccent,
 child:Text('B',style: TextStyle(fontSize: 50,),),),),
 Container(

 child: CircleAvatar(backgroundColor: Colors.green,
 child:Text('C',style: TextStyle(fontSize: 50,),),),),
 Container(

 child: CircleAvatar(backgroundColor: Colors.yellow,
 child:Text('D',style: TextStyle(fontSize: 50,),),),),

 ], )


body:
```

```
GridView.count(crossAxisCount: 2,

children: [
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.cyan,
child:Text('A',style: TextStyle(fontSize: 50,),),),),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.pinkAccent,
child:Text('B',style: TextStyle(fontSize: 50,),),),),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.green,
child:Text('C',style: TextStyle(fontSize: 50,),),),),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.yellow,
child:Text('D',style: TextStyle(fontSize: 50,),),),),
```




```
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.teal,
child:Text('E',style: TextStyle(fontSize: 50,),),),),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.purpleAccent,
child:Text('F',style: TextStyle(fontSize: 50,),),),),
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.brown,
child:Text('G',style: TextStyle(fontSize: 50,),),),),
```

```
),
Padding(
padding: const EdgeInsets.all(8.0),
child: Container(

child: CircleAvatar(backgroundColor: Colors.greenAccent,
child:Text('H',style: TextStyle(fontSize: 50,),),),),),
),

], )



body:
      Container(
        child: GridView.count(crossAxisCount: 5,
        crossAxisSpacing:11,
        mainAxisSpacing: 11,
           children: [
            Container(

              child: Container(color: Colors.cyan,
                child:Text('A',style: TextStyle(fontSize: 50,),),),),),
              Container(
```

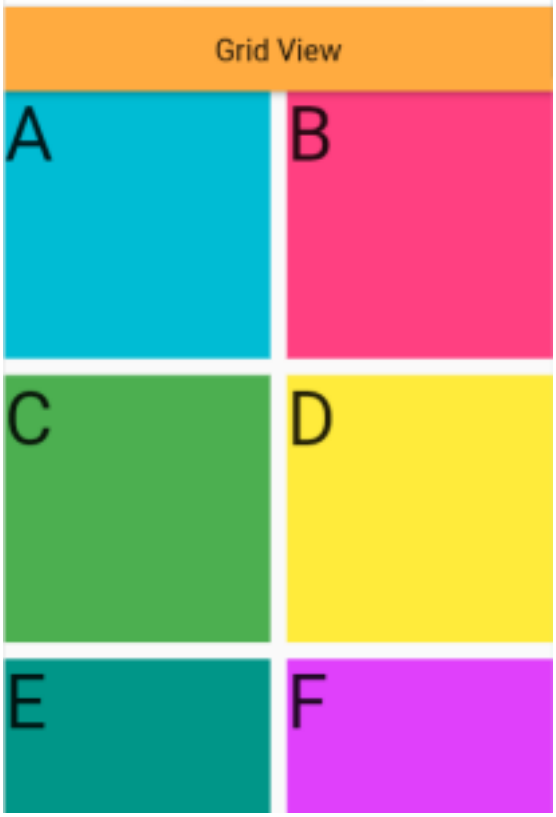


```
              child: Container(color: Colors.pinkAccent,
                child:Text('B',style: TextStyle(fontSize:
                50,),),),),),
              Container(

              child: Container(color: Colors.green,
                child:Text('C',style: TextStyle(fontSize:
                50,),),),),),
              Container(

              child: Container(color: Colors.yellow,
                child:Text('D',style: TextStyle(fontSize:
                50,),),),),),
              Container(

              child: Container(color: Colors.teal,
                child:Text('E',style: TextStyle(fontSize:
                50,),),),),),
              Container(

              child: Container(color: Colors.purpleAccent,
                child:Text('F',style: TextStyle(fontSize: 50,),),),),),
              Container(
```

```
            child: Container(color: Colors.brown,
              child:Text('G',style: TextStyle(fontSize: 50,),),),),),

        ],

        ),
    )

  body:

    Container(
      child:GridView.extent( maxCrossAxisExtent:100,
    crossAxisSpacing:11,
    mainAxisSpacing: 11,
        children: [
          Container(

            child: Container(color: Colors.cyan,
              child:Text('A',style: TextStyle(fontSize: 50,),),),),),
```



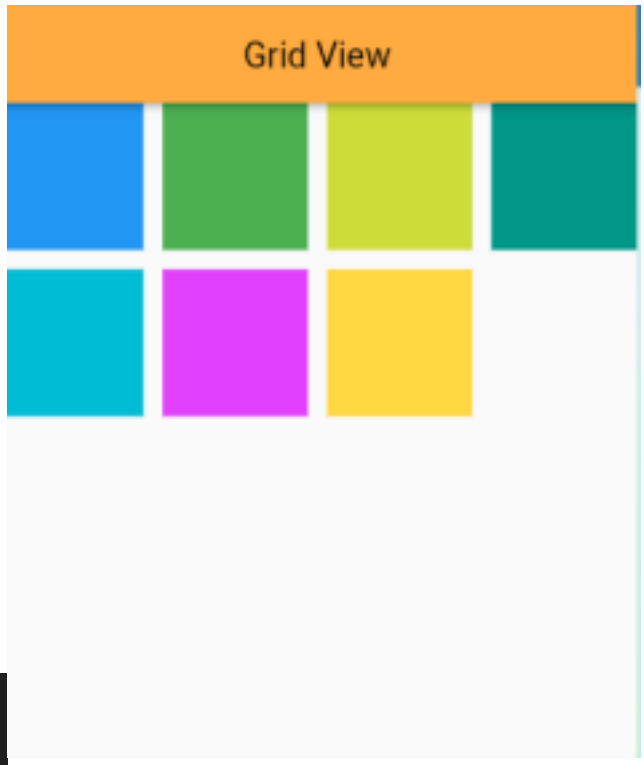
```
          Container(

            child: Container(color:
            Colors.pinkAccent,
              child:Text('B',style: TextStyle(fontSize:
              50,),),),),),
          Container(

            child: Container(color: Colors.green,
              child:Text('C',style: TextStyle(fontSize:
              50,),),),),),
          Container(

            child: Container(color: Colors.yellow,
              child:Text('D',style: TextStyle(fontSize:
              50,),),),),),
          Container(

            child: Container(color: Colors.teal,
              child:Text('E',style: TextStyle(fontSize:
              50,),),),),),
          Container(

            child: Container(color: Colors.purpleAccent,
              child:Text('F',style: TextStyle(fontSize: 50,),),),),),
          Container(

            child: Container(color: Colors.brown,
```



```
                child:Text('G',style: TextStyle(fontSize:
                50,),),),),),

            ],

        ),
        ),

    );


body:

        GridView.builder(itemBuilder:(context,index){

       return Container(color:arrColors[index],);


        },
        itemCount: arrColors.length,
        gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(crossAxisCount: 3
        ), )
```


```
body:

GridView.builder(itemBuilder:(context,index){

 return Container(color:arrColors[index],)
  },
  itemCount: arrColors.length,
 gridDelegate: SliverGridDelegateWithMaxCrossAxisExtent(maxCrossAxisExtent: 100
, crossAxisSpacing:11,
```

mainAxisSpacing: 11),
)



Exercise:

Question 1:

Create a Flutter app that uses the Stack widget to display an image as the background, a semi transparent container overlay, and text in the center. The image should cover the entire screen, and the overlay should have some padding.

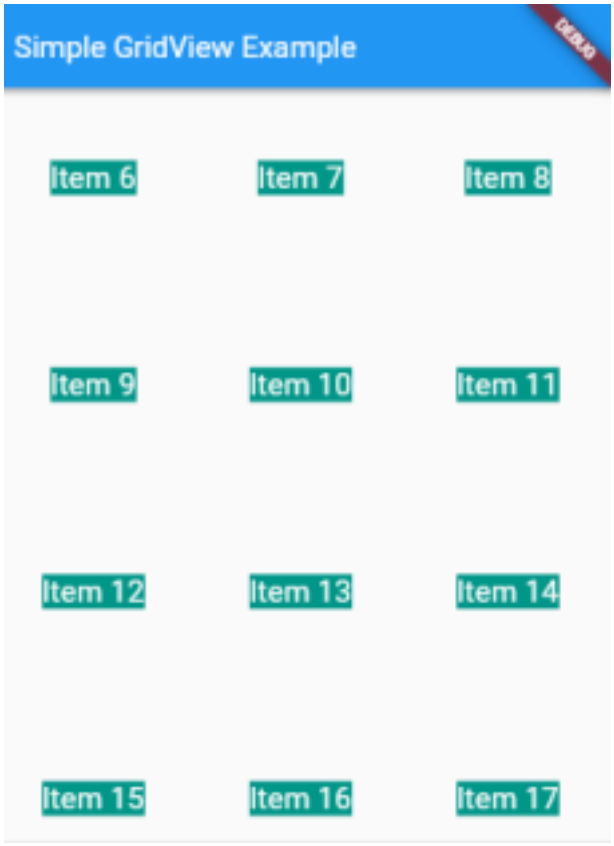
```dart
//Musadiuqe Hussain SE-21031
//Muhammad Asim SE-21045
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Stack(
          children: [
            // Background image
            Container(
              decoration: BoxDecoration(
                image: DecorationImage(
                  image: AssetImage('assets/background.jpg'),
                  fit: BoxFit.cover,
                ),
              ),
            ),
            // Semi-transparent overlay
            Container(
              padding: EdgeInsets.all(16.0),
              color: Colors.black.withOpacity(0.5),
              child: Center(
                child: Text(
                  'Centered Text',
                  style: TextStyle(color: Colors.white, fontSize: 24),
                ),
              ),
            ),
          ],
```

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

Question 2:

Create a Flutter app that displays a GridView with 3 columns. Each grid item should display a centered text indicating its position in the grid.

```
//Musadique Hussain SE-21031
//Muhammad Asim SE-21045
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('GridView with 3 Columns')),
        body: GridView.builder(
          gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 3,
          ),
          itemCount: 20, // Set the number of items you want in the grid
          itemBuilder: (context, index) {
            return Center(
              child: Text(
                'Item $index',
                style: TextStyle(fontSize: 20),
              ),
            );
          },
        ),
      ),
    );
  }
}
```



Question 3:

Create a Flutter app that displays a GridView with 4 columns. The grid items should display numbers from 1 to 100. Each grid item should be a square with a light blue background.

```dart
//Musadique Hussain SE-21031
//Muhammad Asim SE-21045
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('GridView with 4 Columns')),
        body: GridView.builder(
          gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 4,
            childAspectRatio: 1.0, // Ensures each item is a square
          ),
          itemCount: 100,
          itemBuilder: (context, index) {
```

```
            return Container(
              color: Colors.lightBlue,
              child: Center(
                child: Text(
                  '${index + 1}',
                  style: TextStyle(fontSize: 20, color: Colors.white),
                ),
              ),
            );
          },
        ),
      ),
    );
  }
}
```