

Experiment 2:

Aim:- Exploring Flutter Widgets

Theory:

We can split the Flutter widget into two categories:

Visible (Output and Input)

Invisible (Layout and Control)

Visible widget

The visible widgets are related to the user input and output data. Some of the important types of this widget are:

1. Text

A Text widget holds some text to display on the screen. We can align the text widget by using `textAlign` property, and `style` property allow the customization of Text that includes font, font weight, font style, letter spacing, color, and many more. We can use it as like below code snippets.

2. Button

This widget allows you to perform some action on click. Flutter does not allow you to use the Button widget directly; instead, it uses a type of buttons like a `FlatButton` and a `RaisedButton`. We can use it as like below code snippets

3. Image

This widget holds the image which can fetch it from multiple sources like from the asset folder or directly from the URL. It provides many constructors for loading image, which are given below:

- **Image:** It is a generic image loader, which is used by **ImageProvider**.
- **asset:** It load image from your project asset folder.
- **file:** It loads images from the system folder.
- **memory:** It load image from memory.

- **network:** It loads images from the network.

To add an image in the project, you need first to create an assets folder where you keep your images and then add the below line in **pubspec.yaml** file.

Code:-

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

import 'dart:convert';

import 'package:http/http.dart' as http;

import 'package:url_launcher/url_launcher.dart';

class CategoriesPage extends StatefulWidget {

  const CategoriesPage({Key? key}) : super(key: key);

  @override

  _CategoriesPageState createState() => _CategoriesPageState();

}

class _CategoriesPageState extends State<CategoriesPage> {

  final List<String> categories = [

    'Sports',

    'Business',

    'Technology',

    'Health',

    'Science',
```

```
'Entertainment'
```

```
];
```

```
List articles = [];
```

```
String selectedCategory = 'Sports';
```

```
@override
```

```
void initState() {
```

```
    super.initState();
```

```
    fetchNews(selectedCategory);
```

```
}
```

```
Future<void> fetchNews(String category) async {
```

```
    final url = Uri.parse(
```

```
'https://newsapi.org/v2/top-headlines?country=us&category=$category&apiKey=05e80915e833424499779414124e6a23');
```

```
    final response = await http.get(url);
```

```
    if (response.statusCode == 200) {
```

```
        final data = json.decode(response.body);
```

```
        setState(() {
```

```
            articles = data['articles'];
```

```

        selectedCategory = category;

    });

} else {

    throw Exception('Failed to load news');

}

}

void _launchURL(String url) async {

    final Uri uri = Uri.parse(url);

    if (await canLaunchUrl(uri)) {

        await launchUrl(uri, mode: LaunchMode.externalApplication);

    } else {

        throw 'Could not launch $url';

    }

}

@override

Widget build(BuildContext context) {

    return Scaffold(

        backgroundColor: Colors.black, // Dark background

        appBar: AppBar(

```

```
title: const Text(  
  'News Categories',  
  style: TextStyle(fontWeight: FontWeight.bold, color: Colors.white),  
),  
backgroundColor: Colors.deepPurple,  
centerTitle: true,  
),  
body: Column(  
  children: [  
    // Category Buttons  
    SizedBox(  
      height: 50,  
      child: ListView.builder(  
        scrollDirection: Axis.horizontal,  
        itemCount: categories.length,  
        itemBuilder: (context, index) {  
          final category = categories[index];  
          return Padding(  
            padding: const EdgeInsets.symmetric(horizontal: 8.0),
```

```
child: ElevatedButton(  
  
  style: ElevatedButton.styleFrom(  
  
    backgroundColor: selectedCategory == category  
      ? Colors.deepPurpleAccent  
      : Colors.grey[800],  
  
    foregroundColor: Colors.white,  
  
    shape: RoundedRectangleBorder(  
  
      borderRadius: BorderRadius.circular(20),  
  
    ),  
  
  ),  
  
  onPressed: () {  
  
    fetchNews(category.toLowerCase());  
  
  },  
  
  child: Text(category),  
  
),  
  
);  
  
},  
  
),  
  
),
```

```
// News List

Expanded(

  child: articles.isEmpty

    ? const Center(

      child: CircularProgressIndicator(color: Colors.deepPurple))

    : ListView.builder(

      padding: const EdgeInsets.all(10),

      itemCount: articles.length,

      itemBuilder: (context, index) {

        final article = articles[index];

        return NewsCard(

          title: article['title'] ?? 'No Title',

          source: article['source']['name'] ?? 'Unknown Source',

          date: article['publishedAt']?.substring(0, 10) ??

            'Unknown Date',

          description: article['description'] ??

            'No description available.',

          imageUrl: article['urlToImage'],
```

```
        url: article['url'],

        onReadMore: () => _launchURL(article['url']),

    );

},

),

],

),

);

}

}
```

// Reusable News Card Component

```
class NewsCard extends StatelessWidget {

    final String title;

    final String source;

    final String date;

    final String description;

    final String? imageUrl;
```



```
final String? url;
```

```
final VoidCallback onReadMore;
```

```
const NewsCard({
```

```
  Key? key,
```

```
  required this.title,
```

```
  required this.source,
```

```
  required this.date,
```

```
  required this.description,
```

```
  this.imageUrl,
```

```
  this.url,
```

```
  required this.onReadMore,
```

```
}) : super(key: key);
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
  return Card(
```

```
    color: Colors.grey[900], // Dark card background
```

```
    shape: RoundedRectangleBorder(borderRadius: BorderRadius.circular(15)),
```

```
elevation: 4,  
  
margin: const EdgeInsets.symmetric(vertical: 10),  
  
child: Column(  
  
  crossAxisAlignment: CrossAxisAlignment.start,  
  
  children: [  
  
    if (imageUrl != null)  
  
      ClipRRect(  
  
        borderRadius: const BorderRadius.only(  
  
          topLeft: Radius.circular(15), topRight: Radius.circular(15)),  
  
        child: Image.network(  
  
          imageUrl!,  
  
          height: 200,  
  
          width: double.infinity,  
  
          fit: BoxFit.cover,  
  
          errorBuilder: (context, error, stackTrace) => Image.asset(  
  
            'assets/news_placeholder.png',  
  
            height: 200,  
  
            width: double.infinity,  
  
            fit: BoxFit.cover,
```

```
    ),  
    ),  
    ),  
    Padding(  
      padding: const EdgeInsets.all(12),  
      child: Column(  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
          Text(  
            title,  
            style: const TextStyle(  
              fontSize: 18,  
              fontWeight: FontWeight.bold,  
              color: Colors.white),  
          ),  
          const SizedBox(height: 5),  
          Text(  
            '$source • $date',  
            style: const TextStyle(fontSize: 14, color: Colors.white70),
```

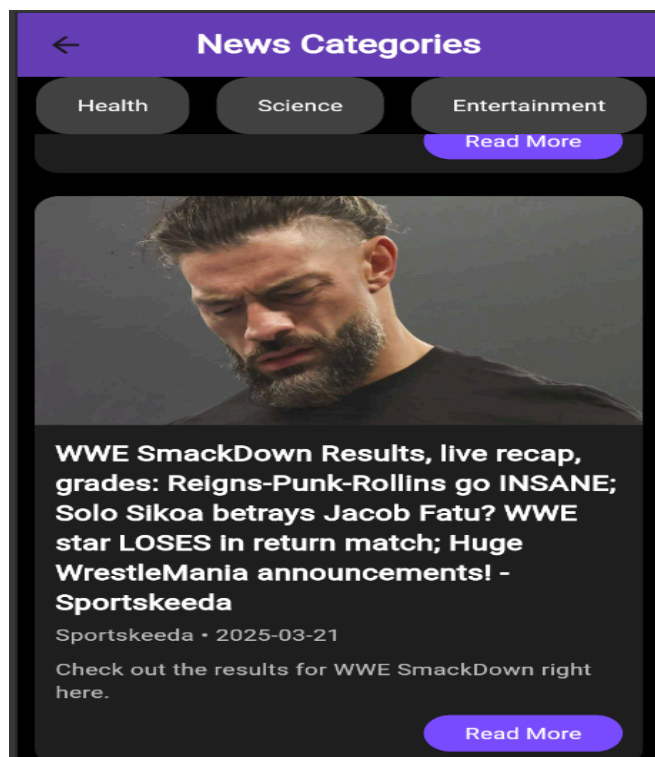
```
),  
  
const SizedBox(height: 10),  
  
Text(  
  
  description,  
  
  maxLines: 3,  
  
  overflow: TextOverflow.ellipsis,  
  
  style: const TextStyle(fontSize: 14, color: Colors.white70),  
  
),  
  
const SizedBox(height: 10),  
  
Align(  
  
  alignment: Alignment.centerRight,  
  
  child: ElevatedButton(  
  
    style: ElevatedButton.styleFrom(  
  
      backgroundColor: Colors.deepPurpleAccent,  
  
      foregroundColor: Colors.white,  
  
    ),  
  
    onPressed: onReadMore,  
  
    child: const Text('Read More'),  
  
  ),
```

```

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

```

Output:-



Here in my NewsEdge Appilcation i have taken images from the NewsAPI which fetches images from the real time news and if no image is available then I have used a placeholder image. There are buttons like Read More which on clicking will go to the expanded news sections. Also I have added some text from myside and taken texts from the NewsAPI.

Github Link:-

https://github.com/Ishan2611/News_Edge/commit/2528bb32429a8c9669175bbb3cadbe8856afed97

Conclusion:-

In conclusion, this experiment provided valuable insights into the core Flutter widgets used for building user interfaces. By exploring visible widgets such as Text, Button, and Image, we learned how to present and interact with data on the screen. Each widget offers customizable properties that enhance user experience and interface design. Understanding how to load images from various sources and implement interactive buttons is essential for any Flutter application. This experiment also emphasized the importance of organizing assets properly and configuring the pubspec.yaml file correctly. Overall, mastering these basic widgets lays a strong foundation for developing functional and visually appealing Flutter apps.