

Experiment 2: Exploring Flutter Widgets

Aim:

The aim of Experiment 2 is to explore various Flutter widgets and understand their usage in building user interfaces for mobile applications.

Theory:

Flutter widgets are the basic building blocks of a Flutter application's user interface. They are responsible for rendering everything that users see and interact with on the screen. Widgets in Flutter can be categorized into two main types: StatelessWidget and StatefulWidget.

StatelessWidget:

- A StatelessWidget is a widget that does not have any mutable state.
- Once created, its properties cannot change.
- Examples include Text, Icon, Button, etc.
- They are typically used for displaying static content that does not change over time.

StatefulWidget:

- A StatefulWidget is a widget that can maintain state and can be updated over time.
- It consists of two classes: one for the widget itself and one for the mutable state.
- Examples include TextField, CheckBox, Slider, etc.
- They are used for interactive elements that require changes in response to user input or other events.

Code :

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

void main() {
  runApp(FootballTournamentApp());
}

class FootballTournamentApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Football Tournament',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: TeamListScreen(),
    );
  }
}

class TeamListScreen extends StatelessWidget {
  final List<String> teams = [
    'Team A',
    'Team B',
    'Team C',
    'Team D',
    'Team E',
    'Team F',
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Teams'),
      ),
      body: ListView.builder(
        itemCount: teams.length,
        itemBuilder: (context, index) {
          return ListTile(
            title: Text(teams[index]),
            onTap: () {
              // Navigate to team details screen
              Navigator.push(
```

```
        context,
        MaterialPageRoute(
          builder: (context) => TeamDetailsScreen(teamName: teams[index]),
        ),
      );
    },
  );
},
),
);
}
}
```

```
class TeamDetailsScreen extends StatelessWidget {
  final String teamName;
```

```
  TeamDetailsScreen({required this.teamName});
```

```
  @override
```

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

```
    return Scaffold(
```

```
      appBar: AppBar(
```

```
        title: Text(teamName),
```

```
      ),
```

```
      body: Center(
```

```
        child: Text('Details of $teamName'),
```

```
      ),
```

```
    );
```

```
  }
```

```
}
```



Conclusion :

we gained insight into constructing dynamic user interfaces and facilitating navigation between different screens. Further enhancements could include incorporating additional widgets for interactive elements, styling, and data visualization to enrich the app's functionality and user experience.