Name : Pranav Chindage Class : D15B Roll no. : 10

Experiment 4 : Form using Flutter Widgets

Aim:

The aim of Experiment 4 is to understand the implementation of forms in Flutter using various widgets to collect user input efficiently and validate data for the Football Tournament app.

Theory:

TextFormField Widget:

- TextFormField widget in Flutter is used to create text input fields within forms.
- It allows users to enter text data and provides various properties for customization, such as decoration, validator, and controller.

Validation:

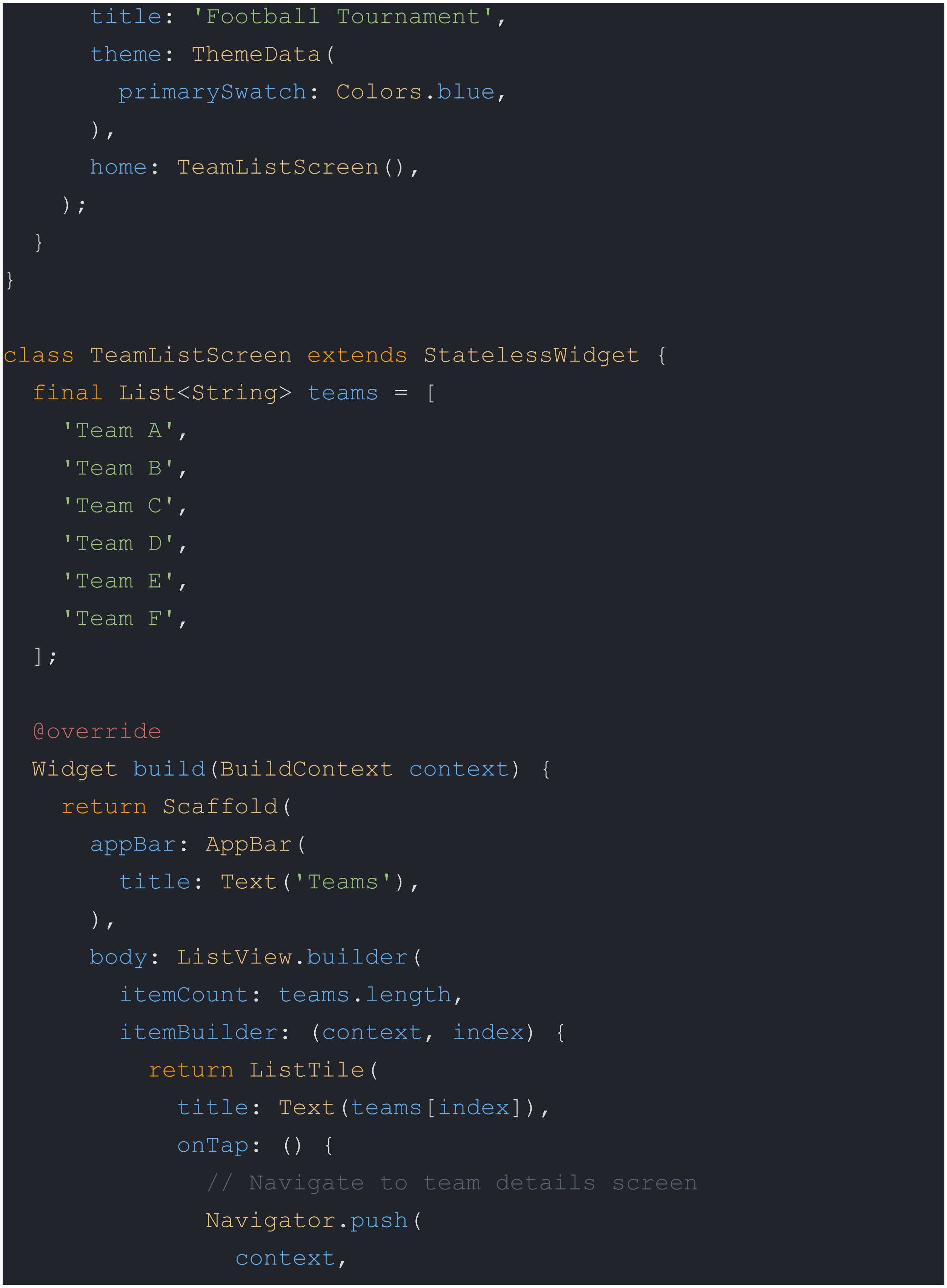
- Flutter provides built-in mechanisms for validating user input in forms.
- Validators can be applied to TextFormField widgets to ensure the entered data meets specific criteria, such as required fields, minimum/maximum length, or pattern matching.

Form Widget:

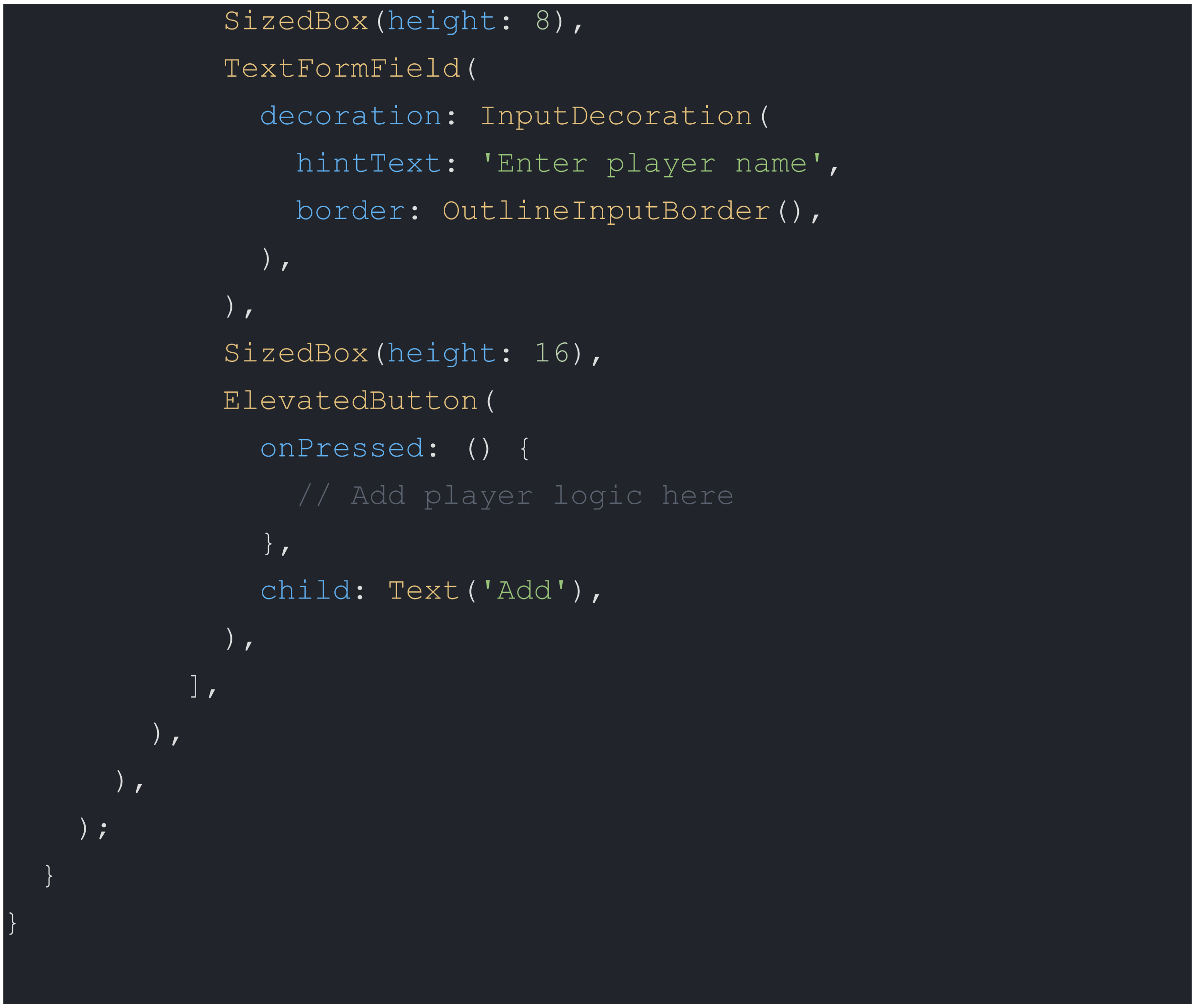
- Form widget in Flutter is used to group multiple form fields together.
- It manages the form's state, including validation, submission, and resetting.

Code :

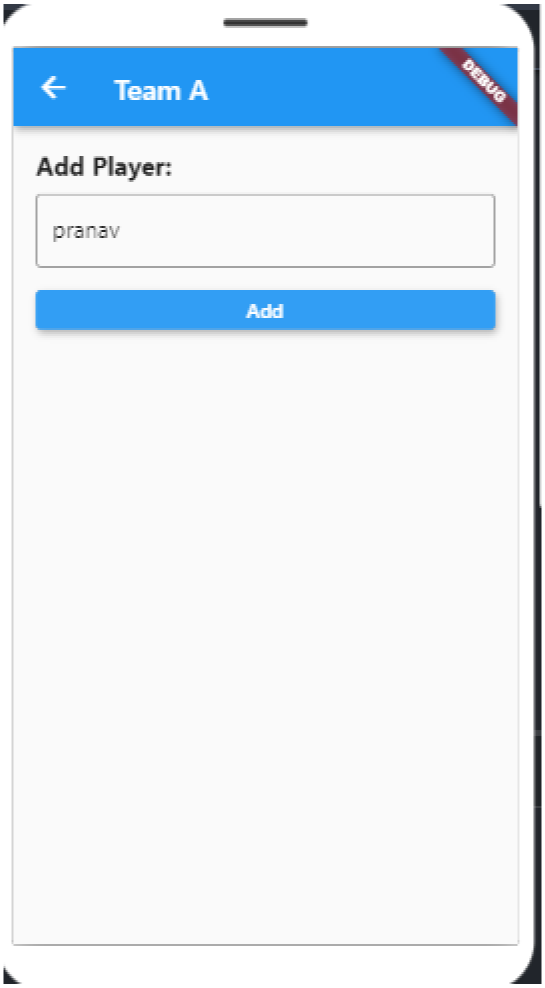
```
import 'package:flutter/material.dart';

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: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.stretch,
          children: [
            Text(
              'Add Player:',
              style: TextStyle(fontSize: 18, fontWeight:
FontWeight.bold),
            ),
```



```
            SizedBox(height: 8),
            TextFormField(
              decoration: InputDecoration(
                hintText: 'Enter player name',
                border: OutlineInputBorder(),
              ),
            ),
            SizedBox(height: 16),
            ElevatedButton(
              onPressed: () {
                // Add player logic here
              },
              child: Text('Add'),
            ),
          ],
        ),
      ),
    );
  }
}
```



pranav

Conclusion :

Integrating forms into the app enhances its functionality and usability, enabling seamless interaction with users and improving data management capabilities. Going forward, further enhancements could include implementing form submission logic and integrating additional validation rules to refine the user experience.