

Name : Vivek Gupta

Div : D15B

Roll No : 19

MPL Practical 04

Aim: To create an interactive Form using form widget

Theory:

Forms are an essential part of mobile applications, allowing users to input and submit data. In Flutter, the `Form` widget provides an organized way to collect and validate user input. It works with `TextFormField` and `DropdownButtonFormField` to ensure structured data entry. The `GlobalKey<FormState>` is used to manage form validation and submission.

Implementation in Code

In our implementation, we created an interactive form to allow users to create a sports team. The key components include:

- Form Widget: Wraps input fields for validation.
- TextFormField: Captures the team name with validation.
- DropdownButtonFormField: Allows users to select a sport.
- Validation: Ensures fields are not left empty.
- Snackbar Feedback: Displays a success message when the form is submitted.

CODE:

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

```
void main() {  
  runApp(TeamFinderApp());  
}
```

```
class TeamFinderApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      title: 'Team Finder',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
        fontFamily: 'Poppins', // Custom font  
        colorScheme: ColorScheme.fromSwatch().copyWith(  
          primary: Colors.blue[900],  
          secondary: Colors.blue[700],  
        ),  
        visualDensity: VisualDensity.adaptivePlatformDensity,  
      ),  
    );  
  }  
}
```

```

        home: CreateTeamScreen(), // Directly opening the form screen
    );
}
}

class CreateTeamScreen extends StatefulWidget {
  @override
  _CreateTeamScreenState createState() => _CreateTeamScreenState();
}

class _CreateTeamScreenState extends State<CreateTeamScreen> {
  final _formKey = GlobalKey<FormState>(); // Form key for validation
  String? _teamName;
  String? _selectedSport;
  List<String> sports = ['Football', 'Basketball', 'Cricket', 'Tennis'];

  void _submitForm() {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Team $_teamName for $_selectedSport created!')),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Create a Team'),
        leading: Icon(Icons.sports), // Icon in AppBar
      ),
      body: Padding(
        padding: EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
              Text(
                'Create Your Team',

```

```

        style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold, color: Colors.blue[900]),
      ),
      SizedBox(height: 10),
      TextFormField(
        decoration: InputDecoration(
          labelText: 'Team Name',
          prefixIcon: Icon(Icons.group, color: Colors.blue[900]),
          border: OutlineInputBorder(),
        ),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) return 'Please enter a team name';
        return null;
      },
      onSave: (value) => _teamName = value,
    ),
    SizedBox(height: 10),
    DropdownButtonFormField<String>(
      decoration: InputDecoration(
        labelText: 'Select Sport',
        prefixIcon: Icon(Icons.sports_soccer, color: Colors.blue[900]),
        border: OutlineInputBorder(),
      ),
    ),
    items: sports.map((sport) {
      return DropdownMenuItem(
        value: sport,
        child: Text(sport),
      );
    }).toList(),
    validator: (value) => value == null ? 'Please select a sport' : null,
    onChanged: (value) {
      setState(() {
        _selectedSport = value;
      });
    },
  ),
  SizedBox(height: 20),
  Center(
    child: ElevatedButton.icon(
      onPressed: _submitForm,
      icon: Icon(Icons.check),
      label: Text('Create Team'),
    ),
  ),

```

```

style: ElevatedButton.styleFrom(
  backgroundColor: Colors.lightBlue[200],
  padding: EdgeInsets.symmetric(horizontal: 30, vertical: 15),
  textStyle: TextStyle(fontSize: 20),
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(10),
  ),
),
),
),
),
],
),
),
),
);
}
}

```

OUTPUT:

The image displays two side-by-side screenshots of a mobile application interface for creating a sports team. Both screens have a title bar with a home icon and the text 'Create a Team'. Below the title bar is a section header 'Create Your Team'. The form consists of two input fields: 'Team Name' with a person icon and 'Select Sport' with a soccer ball icon. A blue button with a checkmark and the text 'Create Team' is positioned below the inputs. In the left screenshot, the 'Team Name' field is empty and the 'Select Sport' dropdown is closed. In the right screenshot, the 'Team Name' field contains the text 'Prevailers' and the 'Select Sport' dropdown is open, showing 'Cricket' as the selected option.

Conclusion:

In this experiment, we successfully implemented an interactive form using the `Form` widget, allowing users to create a sports team with validation and user feedback. Initially, we faced issues with form validation not triggering properly, but we resolved it by ensuring that `GlobalKey<FormState>` was correctly linked and validation checks were properly implemented.