Name: Mitesh A. Dalvi **Roll no.: 11**
Class: D15B

MAD Experiment 4

Aim: To create an interactive form using form widget.

Theory:

The Form widget in Flutter is a fundamental widget for building forms. It provides a way to group multiple form fields together, perform validation on those fields, and manage their state. In this article, we are going to implement the Form widget and explore some properties and Methods of it. A sample video is given below to get an idea about what we are going to do in this article.

Some Properties of Form Widget

- **key:** A GlobalKey that uniquely identifies the Form. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- **child:** The child widget that contains the form fields. Typically, this is a Column, ListView, or another widget that allows you to arrange the form fields vertically.
- **autovalidateMode:** An enum that specifies when the form should automatically validate its fields.

Some Methods of Form Widget

- **validate():** This method is used to trigger the validation of all the form fields within the Form. It returns true if all fields are valid, otherwise false. You can use it to check the overall validity of the form before submitting it.
- **save():** This method is used to save the current values of all form fields. It invokes the onSaved callback for each field. Typically, this method is called after validation succeeds.
- **reset():** Resets the form to its initial state, clearing any user-entered data.
- **currentState:** A getter that returns the current FormState associated with the Form.

Program:

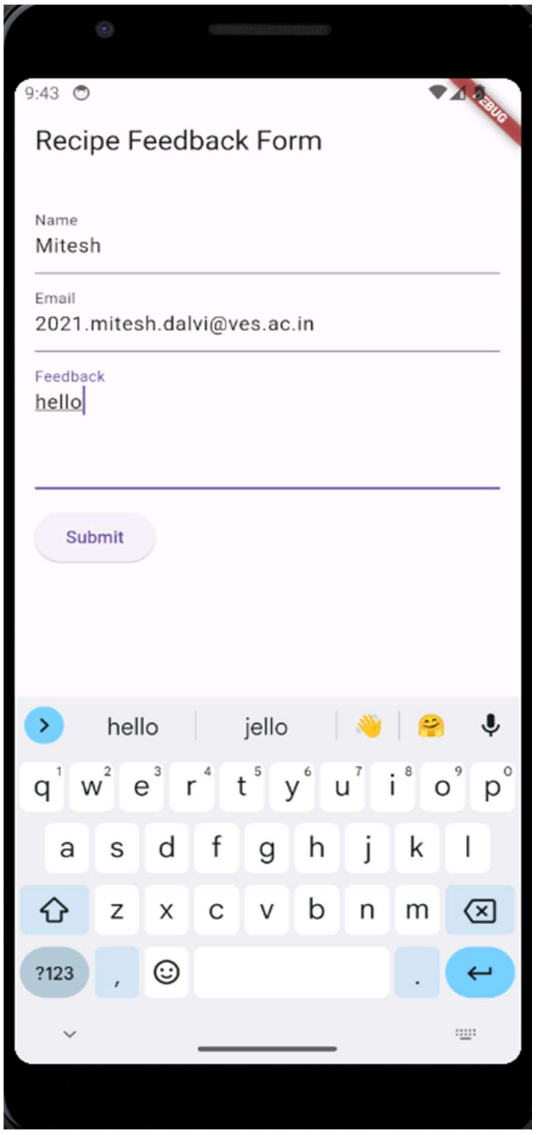
This code creates a basic feedback form with fields for name, email, and feedback. The form validates the inputs and prints them to the console when the submit button is pressed.

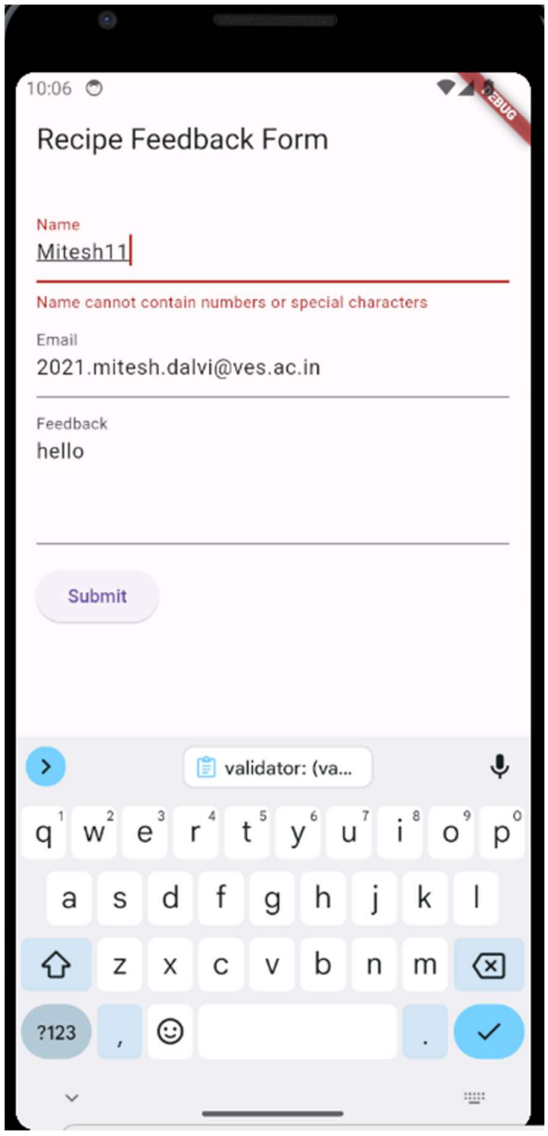
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Recipe Feedback Form',
      home: FeedbackForm(),
    );
  }
}

class FeedbackForm extends StatefulWidget {
  @override
  _FeedbackFormState createState() => _FeedbackFormState();
}

class _FeedbackFormState extends State<FeedbackForm> {
  final _formKey = GlobalKey<FormState>();
  String _name = '';
  String _email = '';
  String _feedback = '';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Recipe Feedback Form'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
```

```
child: Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: <Widget>[
    TextFormField(
      decoration: InputDecoration(labelText: 'Name'),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your name';
        }
        if (RegExp(r'[0-9!@#<>?":_`~;[\]\\|=+)(*&^%$£"/]')
            .hasMatch(value)) {
          return 'Name cannot contain numbers or special characters';
        }
        return null;
      },
      onSaved: (value) {
        _name = value!;
      },
    ),
    TextFormField(
      decoration: InputDecoration(labelText: 'Email'),
      validator: (value) {
        if (value!.isEmpty || !value.contains('@')) {
          return 'Please enter a valid email address';
        }
        return null;
      },
      onSaved: (value) {
        _email = value!;
      },
    ),
    TextFormField(
      decoration: InputDecoration(labelText: 'Feedback'),
      validator: (value) {
        if (value!.isEmpty) {
          return 'Please enter your feedback';
        }
        return null;
      },
      onSaved: (value) {
        _feedback = value!;
      },
      maxLines: 3,
```

```
            ),
            Padding(
              padding: const EdgeInsets.symmetric(vertical: 16.0),
              child: ElevatedButton(
                onPressed: () {
                  if (_formKey.currentState!.validate()) {
                    _formKey.currentState!.save();
                    // Process the data, for example, send it to an API
                    // or save it locally
                    print('Name: $_name');
                    print('Email: $_email');
                    print('Feedback: $_feedback');
                    // Clear the form
                    _formKey.currentState!.reset();
                  }
                },
                child: Text('Submit'),
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

Output:





In console:

```
D/EGL_emulation(15365): app_time_stats: avg=494.12ms min=478.80ms max=502.57ms count=3
I/flutter (15365): Name: Mitesh
I/flutter (15365): Email: 2021.mitesh.dalvi@ves.ac.in
I/flutter (15365): Feedback: Hello
```

Conclusion: In this experiment, we have learnt how to use form widget in our app in Flutter. We have successfully created a form which takes input as email, name and feedback for the recipes in our Recipe App. The form validation ensured that users entered valid information before submitting. The form submission process was implemented to print the feedback data to the console.