

EXPERIMENT - 04

AIM : To create an interactive Form using a form widget.

THEORY :

Flutter provides a Form widget to create a form. The form widget acts as a container, which allows us to group and validate the multiple form fields. When you create a form, it is necessary to provide the GlobalKey. This key uniquely identifies the form and allows you to do any validation in the form fields.

The form widget uses child widget TextFormField to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur

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 onSave 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.**

CODE :

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

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

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text('Flutter Form')        ),  
        body: Center(  
          child: Form(  
            key: GlobalKey(),  
            child: Column(  
              children: [  
                TextFormField(  
                  decoration: InputDecoration(  
                    labelText: 'Name',  
                    border: OutlineInputBorder()  
                  ),  
                TextFormField(  
                  decoration: InputDecoration(  
                    labelText: 'Email',  
                    border: OutlineInputBorder()  
                  ),  
                TextFormField(  
                  decoration: InputDecoration(  
                    labelText: 'Password',  
                    border: OutlineInputBorder()  
                  ),  
                TextFormField(  
                  decoration: InputDecoration(  
                    labelText: 'Confirm Password',  
                    border: OutlineInputBorder()  
                  ),  
              ],  
            ),  
          ),  
        ),  
      ),  
    );  
  }  
}
```

```

        title: Text('Experiment - 4 Forms'),
      ),
      body: MyForm(),
    ),
  );
}
}

class MyForm extends StatefulWidget {
  @override
  _MyFormState createState() =>
    _MyFormState();
}

class _MyFormState extends State<MyForm> {
  final _formKey = GlobalKey<FormState>();
  String _name = "";
  String _email = "";
  String _selectedGender = 'Male';
  bool _subscribeToNewsletter = false;

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            TextFormField(
              decoration: InputDecoration(labelText:
'Username'),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your name';
                }
                return null;
              },
              onChanged: (value) {
                _name = value!;
              },
            ),

```

```

            SizedBox(height: 16),
            TextFormField(
              decoration: InputDecoration(labelText:
'Email'),
              validator: (value) {
                if (value == null || value.isEmpty ||
!value.contains('@')) {
                  return 'Please enter a valid email
address';
                }
                return null;
              },
              onSave: (value) {
                _email = value!;
              },
            ),
            SizedBox(height: 16),
            DropdownButtonFormField<String>(
              value: _selectedGender,
              items: ['Male', 'Female', 'Prefer Not to
say']
                .map((gender) =>
DropdownMenuItem(
                  value: gender,
                  child: Text(gender),
                ))
                .toList(),
              onChanged: (value) {
                setState(() {
                  _selectedGender = value!;
                });
              },
              decoration: InputDecoration(labelText:
'Gender'),
            ),
            SizedBox(height: 16),
            Row(
              children: [
                Checkbox(
                  value: _subscribeToNewsletter,
                  onChanged: (value) {
                    setState(() {
                      _subscribeToNewsletter = value ??
false;
                    });
                  },
                ),

```

```

        Text('I agree to terms and Conditions'),
      ],
    ),
    SizedBox(height: 26),
    ElevatedButton(
      onPressed: () {
        if (_formKey.currentState!.validate()) {
          _formKey.currentState!.save();
          // Process the form data, e.g., send it
to a server
          print('Name: $_name');
          print('Email: $_email');
          print('Gender: $_selectedGender');

          print('Agree to terms and Conditions:
$_subscribeToNewsletter');
          print("");
        }
      },
      child: Text('Submit'),
    ),
  ],
),
);
}
}

```

OUTPUT :



Experiment - 4 Forms

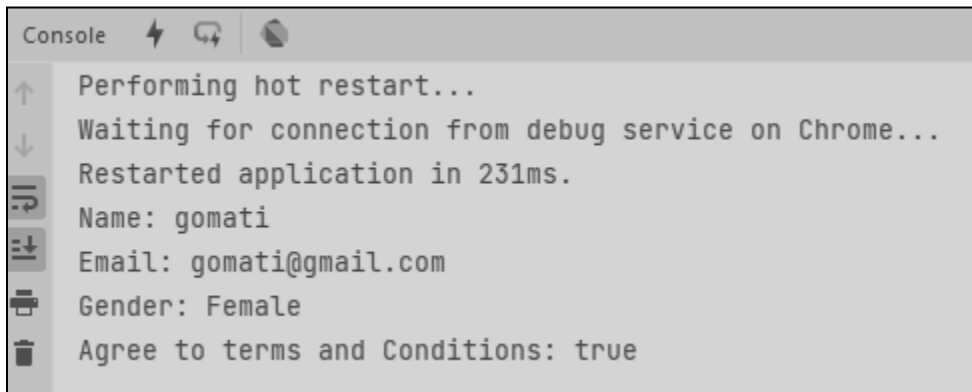
Username
gomati

Email
gomati@gmail.com

Gender
Female

☒ I agree to terms and Conditions

Submit



Console

Performing hot restart...

Waiting for connection from debug service on Chrome...

Restarted application in 231ms.

Name: gomati

Email: gomati@gmail.com

Gender: Female

Agree to terms and Conditions: true