EXPERIMENT NO.1

Experiment No 4 **Exp 4 To create an interactive Form using form widget**	
ROLL NO	**35**
NAME	**Gaurang Milind Mapuskar**
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPED	

Theory:

Flutter Form is a mechanism for capturing and validating user input within a set of input fields. It's constructed using the Form widget, which acts as a container for multiple FormField widgets. FormField widgets represent individual input fields such as TextFormField or DropdownButtonFormField.

GlobalKey:

Key to the functionality of a Flutter Form is the GlobalKey class. GlobalKey is a special type of key that allows access to the state of a widget from anywhere in the widget tree. In the case of forms, GlobalKey is used to uniquely identify the form and access its state, such as validating and submitting the form data.

TextFormField:

TextFormField is a widget used to capture textual input from the user. It provides various properties for customization, such as decoration for styling the input field, keyboardType for specifying the type of input (text, number, email, etc.), and validator for defining validation logic. The validator property takes a function that returns an error message if the input is invalid, or null if the input is valid.

Flutter Validation:

Validation in TextFormField is performed by providing a validator function to the validator property. This function takes the current value of the input field and returns a String error message if the input is invalid, or null if the input is valid. When the form is submitted, each TextFormField's validator function is invoked to check the validity of the input.

Submit Button:

The Submit button in a Flutter form is typically implemented using a button widget, such as ElevatedButton or TextButton. When pressed, the Submit button triggers the form submission process. Before submitting the form, the form's state is checked using the GlobalKey associated with the Form. If the form's state is valid, the form data is

processed or submitted to a backend server. If the form's state is invalid, the user is notified of any validation errors, and the submission is prevented.

Code:

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

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

class PreschoolLearningApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Preschool Learning Form',
      theme: ThemeData(
        primaryColor: Colors.blue,
        hintColor: Colors.orange,
        fontFamily: 'Comic Sans MS', // Example theme for preschool learning
      ),
      home: PreschoolLearningForm(),
    );
  }
}

class PreschoolLearningForm extends StatefulWidget {
  @override
  _PreschoolLearningFormState createState() => _PreschoolLearningFormState();
}

class _PreschoolLearningFormState extends State<PreschoolLearningForm> {
  final _formKey = GlobalKey<FormState>();
  TextEditingController _nameController = TextEditingController();
  TextEditingController _ageController = TextEditingController();

  @override
```

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('Preschool Learning Form'),
    ),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: <Widget>[
            TextFormField(
              controller: _nameController,
              decoration: InputDecoration(
                labelText: 'Name',
              ),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your name';
                }
                return null;
              },
            ),
            TextFormField(
              controller: _ageController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
                labelText: 'Age',
              ),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your age';
                }
                int? age = int.tryParse(value);
                if (age == null || age <= 0) {
                  return 'Please enter a valid age';
```

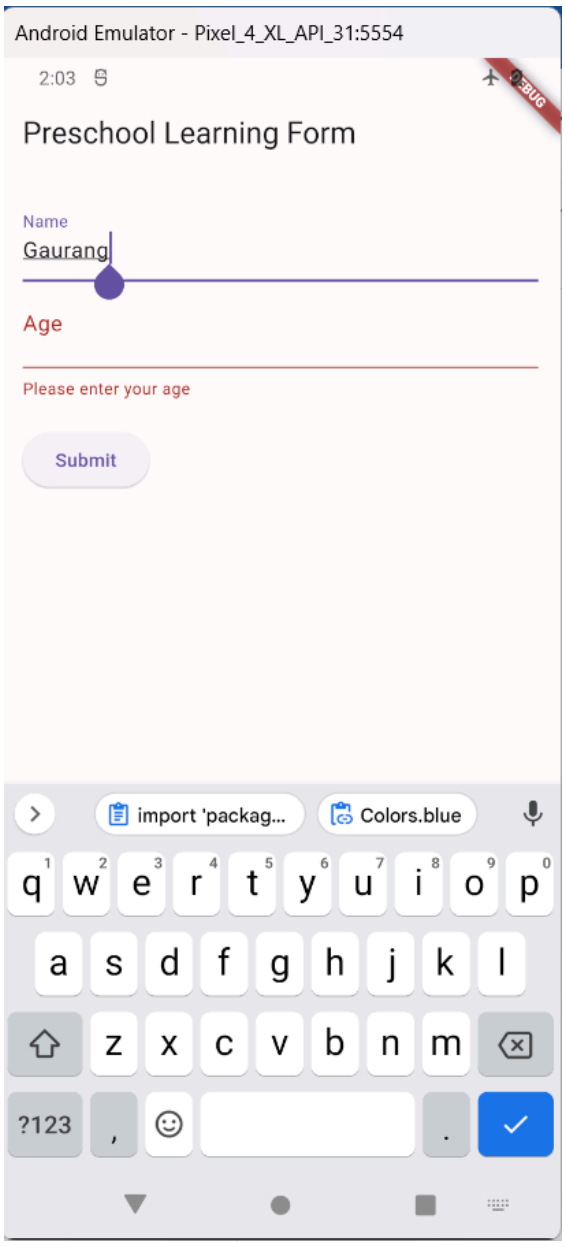
```
              }
              return null;
            },
          ),
          SizedBox(height: 20),
          ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                // Form is valid, submit data
                _submitForm();
              }
            },
            child: Text('Submit'),
          ),
        ],
      ),
    ),
  ),
 );
}

 void _submitForm() {
  String name = _nameController.text;
  int age = int.parse(_ageController.text);

  // Process form data here
  print('Name: $name, Age: $age');
 }

 @override
 void dispose() {
  _nameController.dispose();
  _ageController.dispose();
  super.dispose();
 }
}
```

Output:



Conclusion:

Flutter Form is a powerful mechanism for capturing and validating user input using widgets such as TextFormField, GlobalKey, and button widgets. It allows developers to create interactive forms with custom validation logic and submit user data efficiently.