Experiment No:04

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

Theory:

Enhancing Visual Appeal and Functionality in Flutter Apps with Icons, Images, and Fonts

In Flutter, the Form widget is an essential component for creating interactive user input forms. It enables input validation, data submission, and error handling. Here's an overview of how to utilize the Form widget effectively:

1. **Understanding the Form Widget**

- **Definition:** The Form widget acts as a container that houses multiple form fields, enabling users to input data.
- **State Management:** It manages the state of the form and offers methods for validation and submission.

2. **Building a Form::**

- **Creation:** Wrap your form fields within a Form widget to create a form.
- **Global Key:** Utilize the GlobalKey<FormState> to uniquely identify the form and access its state.

3. **Utilizing Form Fields:**

- **Types:** Various form fields like TextFormField, DropdownButtonFormField, etc., collect user input.
- **Configuration:** Each form field should be associated with a controller (for controlled input) and a validator function to ensure data integrity.

4. **Validation Process:**

- **Purpose:** Validation ensures that user input meets predefined criteria before submission.
- **Implementation:** Specify validation logic using the validator property of form fields. Validators are functions that return an error message if validation fails, or null if the input is valid.

5. **Handling Form Submission:**

- **Triggering Submission:** Submission occurs when the user interacts with a submit button or similar action.

- **Submission Process:** Inside the submission handler, validate the form using the validate method of the FormState. If the form is valid, proceed with the submission logic, such as saving data to a database.

6. **Error Management:**
- **Guiding Users:** If form validation fails, display error messages to guide users in correcting their input.
- **Error Display:** Errors can be shown below each form field or as a general error message at the top of the form.

7. **Cleanup Procedures:**
- **Resource Disposal:** Dispose of form controllers and other resources in the dispose method of the State object to prevent memory leaks.

8. **Additional Features:**
- **Enhancements:** Flutter offers various widgets and utilities to augment forms, such as InputDecoration for customizing form field appearance, FocusNode for managing focus between fields, and SnackBar for displaying feedback messages.

Code:

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

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  static const String _title = 'Flutter Form Validation';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const MyHomePage(),
    );
  }
```

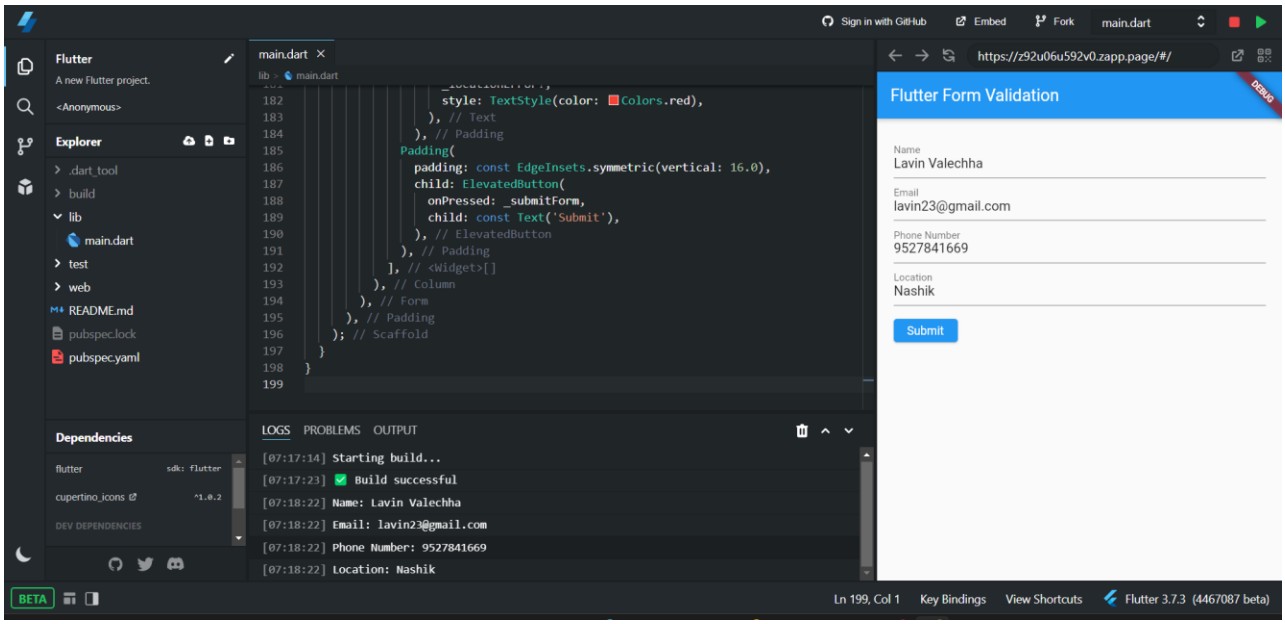
```
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key}) : super(key: key);

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _nameController = TextEditingController();
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _phoneNumberController = TextEditingController();
  final TextEditingController _locationController = TextEditingController();
  String? _nameError;
  String? _emailError;
  String? _phoneNumberError;
  String? _locationError;

  void _submitForm() {
    setState(() {
      _nameError = null;
      _emailError = null;
      _phoneNumberError = null;
      _locationError = null;
    });

    if (_formKey.currentState!.validate()) {
      // Form is validated, proceed with your logic here
      // For example, you can submit the form data
      // You can access form field values using _nameController.text,
_emailController.text, _phoneNumberController.text, and _locationController.text
      // For now, let's just print the values
      print('Name: ${_nameController.text}');
      print('Email: ${_emailController.text}');
      print('Phone Number: ${_phoneNumberController.text}');
      print('Location: ${_locationController.text}');
  }
```

```
  }

  String? _validateName(String? value) {
    if (value == null || value.isEmpty) {
    return 'Name is required';
    }
    return null;
  }

  String? _validateEmail(String? value) {
    if (value == null || value.isEmpty) {
    return 'Email is required';
    } else if (!value.contains('@')) {
      return 'Invalid email format';
    }
    return null;
  }

  String? _validatePhoneNumber(String? value) {
    if (value == null || value.isEmpty) {
      return 'Phone Number is required';
    } else if (value.length != 10) {
      return 'Invalid phone number';
    }
    return null;
  }

  String? _validateLocation(String? value) {
    if (value == null || value.isEmpty) {
    return 'Location is required';
    }
    return null;
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Flutter Form Validation'),
```

```
),
body: Padding(
  padding: const EdgeInsets.all(20.0),
  child: Form(
    key: _formKey,
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: <Widget>[
        TextFormField(
          controller: _nameController,
          decoration: const InputDecoration(
          labelText: 'Name',
          ),
          validator: _validateName,
          onChanged: (_) {
          setState(() {
              _nameError = null;
            });
          },
        ),
        if (_nameError != null)
          Padding(
            padding: const EdgeInsets.only(top: 5),
            child: Text(
              _nameError!,
              style: TextStyle(color: Colors.red),
            ),
          ),
        TextFormField(
          controller: _emailController,
          decoration: const InputDecoration(
          labelText: 'Email',
          ),
          validator: _validateEmail,
          onChanged: (_) {
          setState(() {
              _emailError = null;
            });
          },
```

```
),
if (_emailError != null)
  Padding(
    padding: const EdgeInsets.only(top: 5),
    child: Text(
      _emailError!,
      style: TextStyle(color: Colors.red),
    ),
  ),
TextFormField(
  controller: _phoneNumberController,
  decoration: const InputDecoration(
  labelText: 'Phone Number',
  ),
  validator: _validatePhoneNumber,
  onChanged: (_) {
    setState(() {
      _phoneNumberError = null;
    });
  },
),
if (_phoneNumberError != null)
  Padding(
    padding: const EdgeInsets.only(top: 5),
    child: Text(
      _phoneNumberError!,
      style: TextStyle(color: Colors.red),
    ),
  ),
TextFormField(
  controller: _locationController,
  decoration: const InputDecoration(
  labelText: 'Location',
  ),
  validator: _validateLocation,
  onChanged: (_)  {
  setState(() {
      _locationError = null;
    });
```

```
              },
            ),
            if (_locationError != null)
              Padding(
                padding: const EdgeInsets.only(top: 5),
                child: Text(
                  _locationError!,
                  style: TextStyle(color: Colors.red),
                ),
              ),
            Padding(
              padding: const EdgeInsets.symmetric(vertical: 16.0),
              child: ElevatedButton(
                onPressed: _submitForm,
                child: const Text('Submit'),
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```


Sign in with GitHub
Embed
Fork
main.dart
Flutter
A new Flutter project.
<Anonymous>
Explorer
.dart_tool
build
lib
main.dart
test
web
README.md
pubspec.lock
pubspec.yaml
Dependencies
flutter sdk: flutter
cupertino_icons ^1.0.2
DEV DEPENDENCIES
main.dart
lib > main.dart
style: TextStyle(color: Colors.red),
), // Text
), // Padding
Padding(
padding: const EdgeInsets.symmetric(vertical: 16.0),
child: ElevatedButton(
onPressed: _submitForm,
child: const Text('Submit'),
), // ElevatedButton
), // Padding
], // <Widget>[]
), // Column
), // Form
), // Padding
); // Scaffold
}
}
LOGS PROBLEMS OUTPUT
[07:17:14] Starting build...
[07:17:23] Build successful
[07:18:22] Name: Lavin Valechha
[07:18:22] Email: lavin23@gmail.com
[07:18:22] Phone Number: 9527841669
[07:18:22] Location: Nashik
https://z92u06u592v0.zapp.page/#/
Flutter Form Validation
DEBUG
Name
Lavin Valechha
Email
lavin23@gmail.com
Phone Number
9527841669
Location
Nashik
Submit
BETA
Ln 199, Col 1
Key Bindings
View Shortcuts
Flutter 3.7.3 (4467087 beta)

2.Login Form:

Code:

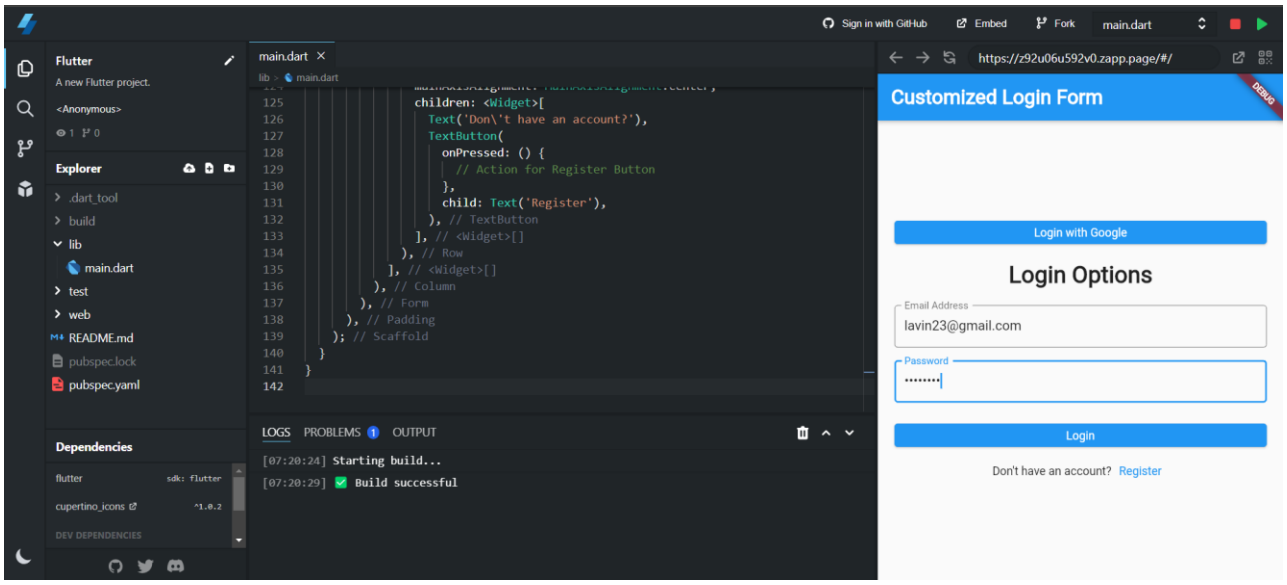
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Login Form',
      theme: ThemeData(
        primaryColor: Colors.green, // Changed primary color to green
        appBarTheme: AppBarTheme(
          titleTextStyle: TextStyle(
            fontSize: 24.0, // Increased AppBar title font size
            fontWeight: FontWeight.bold,
            color:
                Colors.white, // Changed text color of the AppBar title to white
          ),
```

```
        ),
      ),
      home: LoginForm(),
    );
  }
}

class LoginForm extends StatefulWidget {
  @override
  _LoginFormState createState() => _LoginFormState();
}

class _LoginFormState extends State<LoginForm> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  TextEditingController _emailController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  void _submitForm() {
    if (_formKey.currentState != null && _formKey.currentState!.validate()) {
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Login Successful!')),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Customized Login Form'), // Changed AppBar title
      ),
      body: Padding(
```

```
padding: EdgeInsets.all(20.0), // Increased padding
child: Form(
  key: _formKey,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: <Widget>[
      ElevatedButton(
        onPressed: () {
          // Action for Login using Google
        },
        child: Text('Login with Google'),
      ),
      SizedBox(height: 20),
      Text(
        'Login Options',
        style: TextStyle(
            fontSize: 28,
            fontWeight: FontWeight.bold), // Increased font size
        textAlign: TextAlign.center,
      ),
      SizedBox(height: 20),
      TextFormField(
        controller: _emailController,
        decoration: InputDecoration(
        labelText: 'Email Address',
        border: OutlineInputBorder(),
        ),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please enter your email';
          }
          if (!value.contains('@')) {
            return 'Please enter a valid email';
          }
          return null;
        },
      ),
      SizedBox(height: 15), // Increased SizedBox height
```

```
        TextFormField(
          controller: _passwordController,
          obscureText: true,
          decoration: InputDecoration(
            labelText: 'Password',
            border: OutlineInputBorder(),
          ),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please enter your password';
            }
            if (value.length < 8) {
              // Increased minimum password length
              return 'Password must be at least 8 characters long';
            }
            return null;
          },
        ),
        SizedBox(height: 25), // Increased SizedBox height
        ElevatedButton(
          onPressed: _submitForm,
          child: Text('Login'),
        ),
        SizedBox(height: 15), // Increased SizedBox height
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text('Don\'t have an account?'),
            TextButton(
              onPressed: () {
                // Action for Register Button
              },
              child: Text('Register'),
            ),
          ],
        ),
      ],
    ),
  ),
```

```
      ),
    );
  }
}
```



Output:

Conclusion: I have successfully created an interactive Form using form widget in Flutter.