EXPERIMENT NO. 04

Aim: To create an interactive Form using a form widget

Theory:

Steps
1. Create a Form with a GlobalKey.
2. Add a TextFormField with validation logic.
3. Create a button to validate and submit the form.

Form Widget:
The Form widget in Flutter provides a way to manage and validate a group of form fields. It is a container for various form-related widgets.
A form usually consists of multiple form fields, such as text fields, checkboxes, radio buttons, and buttons.

Form Fields:

1. **TextFormField:**

The TextFormField widget is used for single-line text input. It automatically handles validation, error messages, and updating the form state.
Example:

```
TextFormField(
  decoration: InputDecoration(labelText:
  'Username'), validator: (value) {
    if (value == null || value.isEmpty)
      { return 'Please enter your
      username';
    }
    return null;
  },
)
```

2. **DropdownButtonFormField:**

The DropdownButtonFormField widget creates a dropdown menu and automatically manages its state.

Example:

```
DropdownButtonFormField<String
>( value: selectedCountry,
 items: ['USA', 'Canada', 'UK'].map((String
  country) { return
  DropdownMenuItem<String>(
   value: country,
   child: Text(country),
  );
 }).toList(),
 onChanged:
 (value) {
 setState(() {
   selectedCountry = value;
  });
 },
 decoration: InputDecoration(labelText: 'Country'),
)
```

3. **CheckboxFormField:**

The CheckboxFormField widget is used for checkbox input.

Example:

```
CheckboxFormFiel
d( initialValue:
false,
 title: 'Accept Terms and
 Conditions', validator: (value) {
  if (value != true) {
   return 'Please accept the terms and conditions';
  }
  return null;
 },
)
```

Form Validation:
Form validation ensures that the user input meets specific criteria before allowing form submission. It helps maintain data integrity.

Validation is typically done using the validator property of form fields. Validators are functions that return an error message if the input is invalid or null if it's valid.

Code:

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

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: MySignUpForm(),
      theme: ThemeData(
        primaryColor: Colors.teal, // Changed primary color to teal
        colorScheme: ColorScheme.fromSwatch(primarySwatch: Colors.teal), //
Adjusted color scheme
        fontFamily: 'Arial',
      ),
    );
  }
}

class MySignUpForm extends StatefulWidget {
  @override
  _MySignUpFormState createState() => _MySignUpFormState();
}

class _MySignUpFormState extends State<MySignUpForm> {
  final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
  final TextEditingController _nameController = TextEditingController();
```

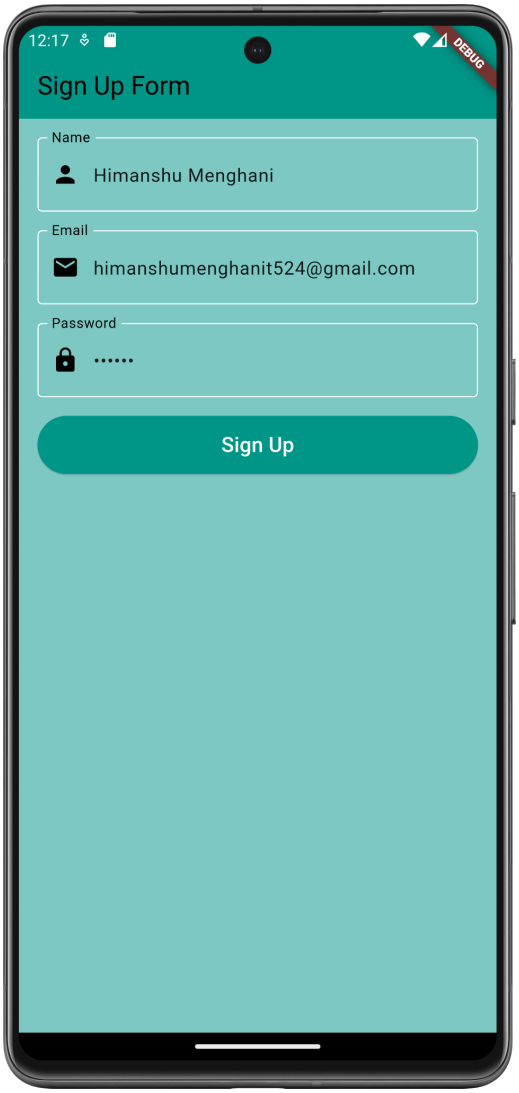
```
String _email = '';
String _password = '';

String? _validateName(String? value) {
  if (value == null || value.isEmpty) {
    return 'Please enter your name';
  }
  return null;
}

String? _validateEmail(String? value) {
  if (value == null || value.isEmpty) {
    return 'Please enter your email';
  } else if (!RegExp(r'^[\w-]+(\.[\w-]+)*@([\w-]+\.)+[a-zA-Z]{2,7}$')
      .hasMatch(value)) {
    return 'Please enter a valid email address';
  }
  return null;
}

String? _validatePassword(String? value) {
  if (value == null || value.isEmpty) {
    return 'Please enter your password';
  } else if (value.length < 6) {
    return 'Password must be at least 6 characters';
  }
  return null;
}

void _submitForm() {
  if (_formKey.currentState?.validate() ?? false) {
    _formKey.currentState?.save();
    _showSignUpCompleteDialog(_nameController.text);
  }
}

void _showSignUpCompleteDialog(String name) {
  showDialog(
    context: context,
```

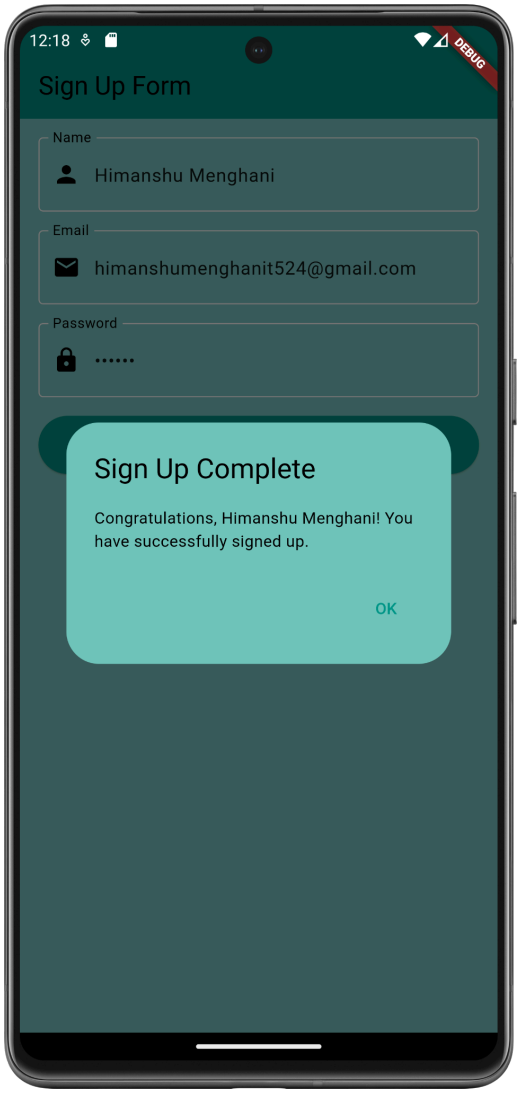
```
      builder: (BuildContext context) {
        return AlertDialog(
          title: Text('Sign Up Complete'),
          content: Text('Congratulations, $name! You have successfully signed up.'),
          actions: <Widget>[
            TextButton(
              onPressed: () {
                Navigator.of(context).pop();
              },
              child: Text('OK'),
            ),
          ],
        );
      },
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Sign Up Form'),
        backgroundColor: Colors.teal, // Changed to teal
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
              TextFormField(
                controller: _nameController,
                decoration: InputDecoration(
                  labelText: 'Name',
                  hintText: 'Enter your name',
                  border: OutlineInputBorder(),
                  prefixIcon: Icon(Icons.person),
                ),
```

```
  style: TextStyle(
   fontSize: 16,
   color: Colors.black87,
  ),
  validator: _validateName,
  onSaved: (value) {
   _nameController.text = value ?? '';
  },
 ),
 SizedBox(height: 16),
 TextFormField(
  decoration: InputDecoration(
   labelText: 'Email',
   hintText: 'Enter your email',
   border: OutlineInputBorder(),
   prefixIcon: Icon(Icons.email),
  ),
  style: TextStyle(
   fontSize: 16,
   color: Colors.black87,
  ),
  validator: _validateEmail,
  onSaved: (value) {
   _email = value ?? '';
  },
 ),
 SizedBox(height: 16),
 TextFormField(
  obscureText: true,
  decoration: InputDecoration(
   labelText: 'Password',
   hintText: 'Enter your password',
   border: OutlineInputBorder(),
   prefixIcon: Icon(Icons.lock),
  ),
  style: TextStyle(
   fontSize: 16,
   color: Colors.black87,
  ),
```

```
              validator: _validatePassword,
              onSaved: (value) {
                _password = value ?? '';
              },
            ),
            SizedBox(height: 16),
            ElevatedButton(
              onPressed: _submitForm,
              child: Text(
                'Sign Up',
                style: TextStyle(
                  fontSize: 18,
                  color: Colors.white,
                ),
              ),
              style: ButtonStyle(
                backgroundColor: MaterialStateProperty.all(Colors.teal), // Changed to
teal
                padding: MaterialStateProperty.all(
                  EdgeInsets.symmetric(vertical: 12),
                ),
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

Output:





Conclusion:

In summary, this experiment has demonstrated the effective utilization of form widgets in Flutter to construct interactive and user-friendly forms within applications. We created a form with a Global field and then added a textform field with validation logic then created a button to validate and submit the form. We can add the form functionality to create our login and sign-up page while creating our project and do user authentication.