MAD Ex 4

Aim: To create an interactive form using the form widget in Flutter.

Theory:

The Form widget in Flutter provides a structured way to handle user input with validation and state management. It contains form fields like:

- Text form fields (TextFormField)
- Dropdown buttons (DropdownButtonFormField)
- Checkbox lists

While working alongside GlobalKey FormState, form validation and submission validation are implemented using the validator property. Form actions like validate(), save(), and reset() ensure efficient input handling. By utilizing these widgets, developers can create user-friendly and responsive forms that enhance data accuracy and user experience.

Importance of using the Form widget:

1) **Ensures data validation** – Prevents incorrect user input.
2) **Manages form state** – Tracks changes and updates fields.
3) **Enhances user experience** – Provides interactive input handling.
4) **Simplifies form submission** – Enables easy data processing and handling.

Conclusion:

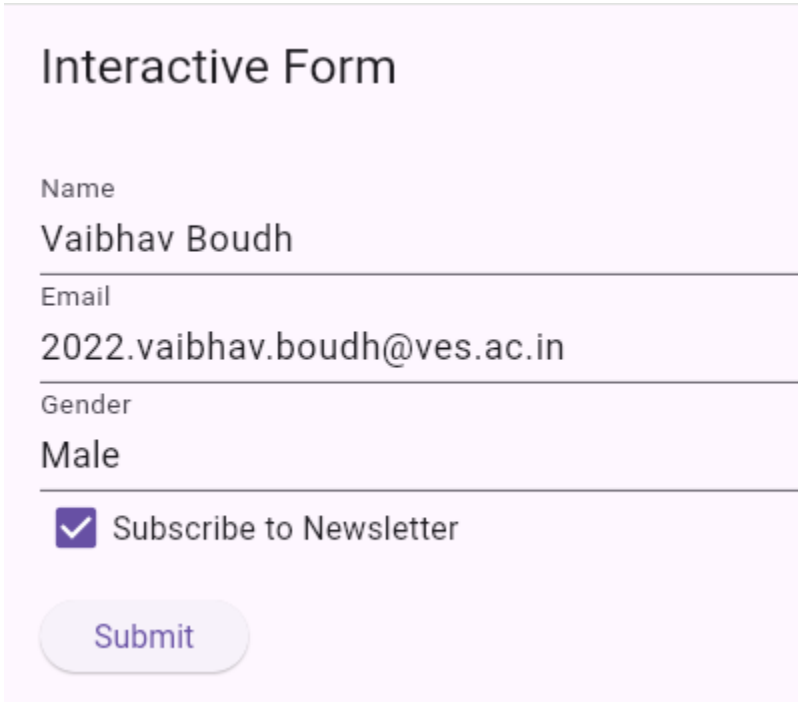
Creating an interactive form using the Form widget in Flutter improves user input handling by providing structured validation and state management. By utilizing various form elements like TextFormField, DropdownButtonFormField, and buttons, developers can design efficient and user-friendly forms for mobile applications.

Code to create an interactive Form using form widget

```
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('Interactive Form'),
        ),
        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: 'Name'),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter your name';
                }
                return null;
```

```
  },
  onSaved: (value) {
    _name = value!;
  },
),
TextFormField(
  decoration: InputDecoration(labelText: 'Email'),
  validator: (value) {
    if (value == null || value.isEmpty || !value.contains('@')) {
      return 'Please enter a valid email address';
    }
    return null;
  },
  onSaved: (value) {
    _email = value!;
  },
),
DropdownButtonFormField<String>(
  value: _selectedGender,
  items: ['Male', 'Female', 'Other']
      .map((gender) => DropdownMenuItem(
            value: gender,
            child: Text(gender),
          ))
      .toList(),
  onChanged: (value) {
    setState(() {
      _selectedGender = value!;
    });
  },
  decoration: InputDecoration(labelText: 'Gender'),
),
Row(
  children: [
    Checkbox(
      value: _subscribeToNewsletter,
      onChanged: (value) {
        setState(() {
          _subscribeToNewsletter = value ?? false;
        });
      },
    ),
    Text('Subscribe to Newsletter'),
  ],
```

```
            ),
            SizedBox(height: 16),
            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('Subscribe to Newsletter: $_subscribeToNewsletter');
                }
              },
              child: Text('Submit'),
            ),
          ],
        ),
      ),
    );
  }
}
```



```
Name: Vaibhav Boudh
Email: 2022.vaibhav.boudh@ves.ac.in
Gender: Male
Subscribe to Newsletter: true
Application finished.
```

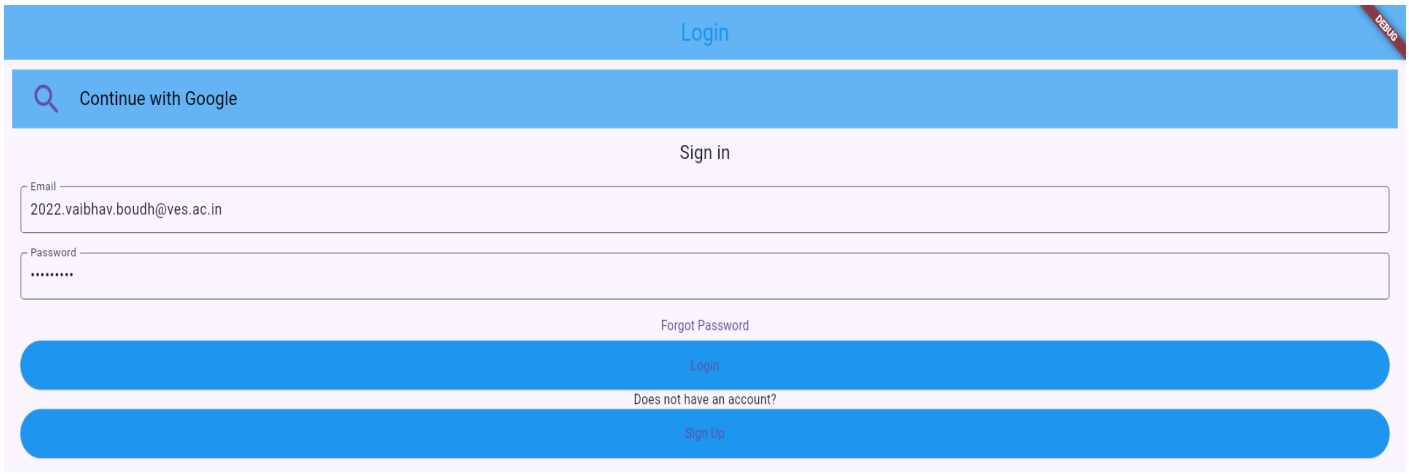
Code to create an interactive Login and sign in page

```
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 = 'Sample App';
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      home: Scaffold(
        appBar: AppBar(
          elevation: 0,
          backgroundColor: Colors.blue[300],
          centerTitle: true,
          title: Text(
            'Login',
            style: TextStyle(
              color: Colors.blue[500],
              fontFamily: 'Roboto',
              fontSize: 24,
            ),
          ),
        ),
        body: const MyStatefulWidget(),
      ),
    );
  }
}
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);
  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(10),
      child: ListView(
        children: <Widget>[
          Container(
```

```
      width: double.infinity,
alignment: Alignment.centerLeft,
      padding: const EdgeInsets.all(10),
      color: Colors.blue[300],
      child: Column(
        children: [
          TextButton(
            onPressed: () {},
            child: Row(
              children: [
                Icon(
                  Icons.search, // Replace with the appropriate Google icon
                  size: 40,
                ),
                SizedBox(width: 20),
                Text(
                  'Continue with Google',
                  style: TextStyle(
                    color: Colors.black,
                    fontFamily: 'Roboto',
                    fontSize: 20,
                  ),
                ),
              ],
            ),
          ),
        ],
      ),
    ),
    Container(
      alignment: Alignment.center,
      padding: const EdgeInsets.all(10),
      child: const Text(
        'Sign in',
        style: TextStyle(fontSize: 20),
      ),
    ),
    Container(
      padding: const EdgeInsets.all(10),
      child: TextField(
        controller: emailController,
        decoration: const InputDecoration(
          border: OutlineInputBorder(),
          labelText: 'Email',
```

```
    ),
  ),
),
Container(
  padding: const EdgeInsets.all(10),
  child: TextField(
    obscureText: true,
    controller: passwordController,
    decoration: const InputDecoration(
      border: OutlineInputBorder(),
      labelText: 'Password',
    ),
  ),
),
TextButton(
  onPressed: () {
    // Forgot password action
  },
  child: const Text('Forgot Password'),
),
Container(
  height: 50,
  padding: const EdgeInsets.fromLTRB(10, 0, 10, 0),
  child: ElevatedButton(
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.blue, // Set the background color to blue
    ),
    child: const Text('Login'),
    onPressed: () {
      print(emailController.text);
      print(passwordController.text);
    },
  ),
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    const Text('Does not have an account?'),
  ],
),
Container(
  height: 50,
  padding: const EdgeInsets.fromLTRB(10, 0, 10, 0),
  child: ElevatedButton(
```

```
            style: ElevatedButton.styleFrom(
              backgroundColor: Colors.blue, // Set the background color to blue
            ),
            child: const Text('Sign Up'),
            onPressed: () {
              print(emailController.text);
              print(passwordController.text);
            },
          ),
        ),
      ],
    ),
  );
 }
}
```

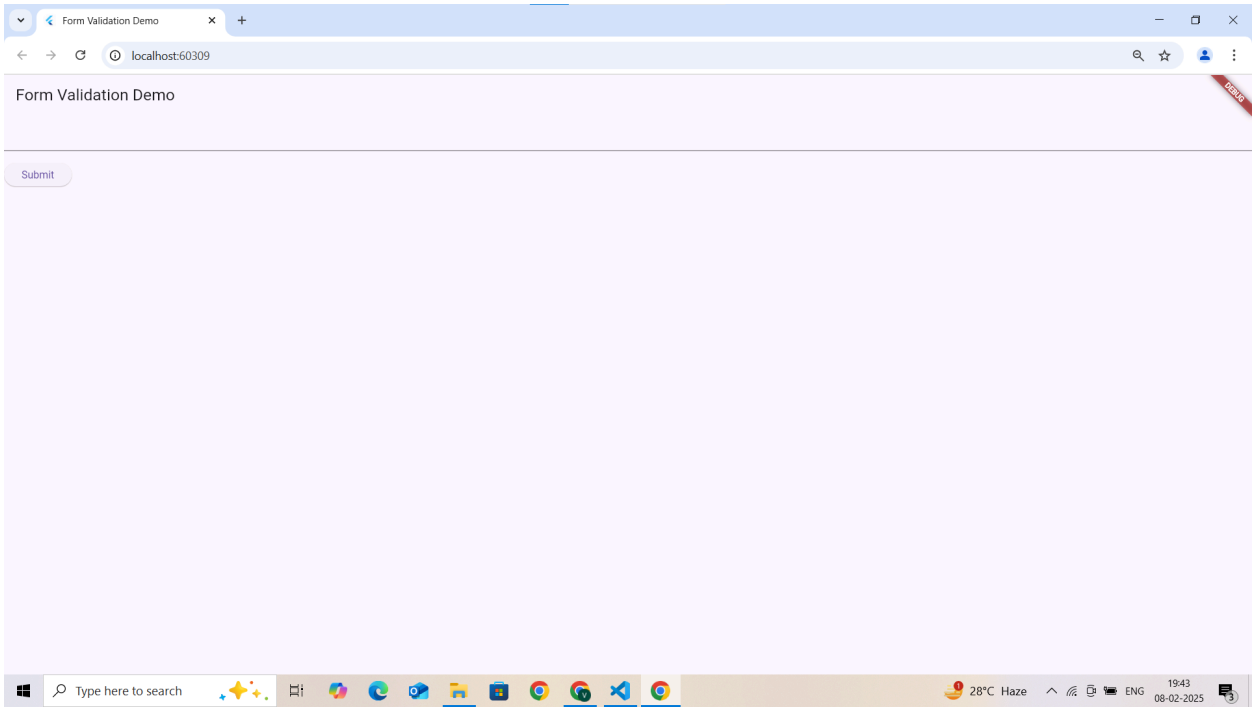


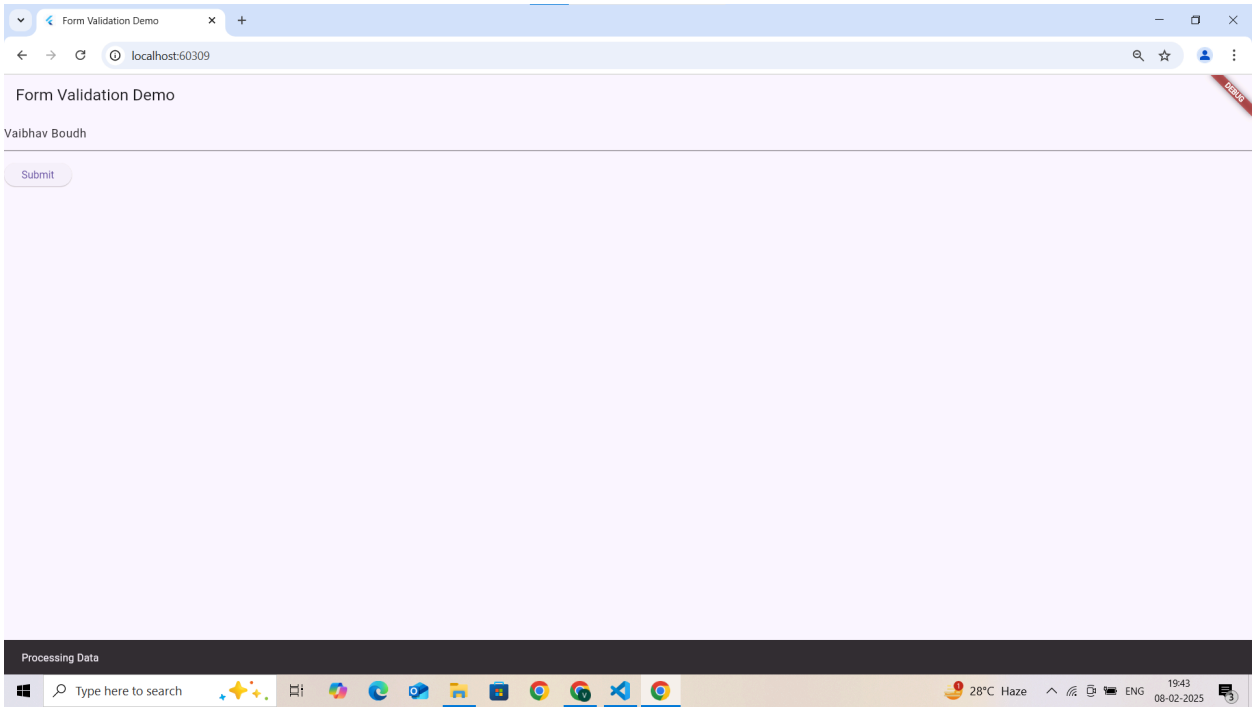

```
A Dart VM Service on Chrome is available at: http://127.0.0.1:60153/lN_NLVXpAAo=
The Flutter DevTools debugger and profiler on Chrome is available at: http://127.0.
2022.vaibhav.boudh@ves.ac.in
147852369
```

Create an interactive Validation page

```
import 'package:flutter/material.dart';
void main() => runApp(const MyApp());
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    const appTitle = 'Form Validation Demo';
    return MaterialApp(
      title: appTitle,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(appTitle),
        ),
        body: const MyCustomForm(),
      ),
    );
  }
}
// Create a Form widget.
class MyCustomForm extends StatefulWidget {
  const MyCustomForm({Key? key}) : super(key: key);
  @override
  MyCustomFormState createState() {
    return MyCustomFormState();
  }
}
// Create a corresponding State class.
class MyCustomFormState extends State<MyCustomForm> {
  // Create a global key that uniquely identifies the Form widget
  // and allows validation of the form.
  final _formKey = GlobalKey<FormState>();
  @override
  Widget build(BuildContext context) {
    // Build a Form widget using the _formKey created above.
    return Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            TextFormField(
              decoration: const InputDecoration(
```

```
              labelText: 'Enter some text',
              border: OutlineInputBorder(),
            ),
            // The validator receives the text that the user has entered.
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter some text';
              }
              return null;
            },
          ),
          const SizedBox(height: 20),
          Center(
            child: ElevatedButton(
              onPressed: () {
                // Validate returns true if the form is valid, or false otherwise.
                if (_formKey.currentState!.validate()) {
                  // If the form is valid, display a snackbar.
                  ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(content: Text('Processing Data')),
                  );
                }
              },
              child: const Text('Submit'),
            ),
          ),
        ],
      ),
    ),
  );
 }
}
```


Form Validation Demo
localhost:60309
Form Validation Demo
DEBUG
Submit
Type here to search
28°C Haze
ENG
19:43
08-02-2025


Form Validation Demo
localhost:60309
Form Validation Demo
DEBUG
Vaibhav Boudh
Submit
Processing Data
Type here to search
28°C Haze
ENG
19:43
08-02-2025