Aim: To create an interactive Form using a form widget

Theory:

In Flutter, forms are a fundamental part of building user interfaces, allowing users to input data and interact with your application. The TextFormField widget is commonly used to create text input fields within a form. Here's an overview of each component you mentioned:

1. **Form**: A Form widget is a container for a group of form fields (such as TextFormField, Checkbox, Radio, etc.). It manages the form's state, validation, and submission.
2. **TextFormField**: A TextFormField widget is used to create a text input field within a form. It provides several properties for customizing the appearance and behavior of the input field, such as decoration (for styling), validator (for input validation), onSaved (for saving the input value), and more.
3. **Button**: In the context of forms, buttons are used to submit or reset the form. You can use the ElevatedButton, TextButton, OutlinedButton, or IconButton widgets to create buttons in Flutter. Typically, a button's onPressed callback is used to handle form submission or other actions.
4. **Validator**: A validator is a function that determines whether the input provided by the user is valid. It is commonly used with form fields like TextFormField to perform validation before submitting the form. The validator function takes the input value as a parameter and returns an error message if the input is invalid, or null if the input is valid.
5. **FormKey**: The FormKey is a unique key that identifies a Form widget in Flutter. It's used to access and interact with the form's state, including validating and submitting the form. You typically create a GlobalKey<FormState> and assign it to the key property of the Form widget.

Code:

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

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

class PomodoroApp extends StatelessWidget {
```

```
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Pomodoro Timer',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: PomodoroTimer(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class PomodoroTimer extends StatefulWidget {
  @override
  _PomodoroTimerState createState() => _PomodoroTimerState();
}

class _PomodoroTimerState extends State<PomodoroTimer> {
  int _minutes = 25;
  int _seconds = 0;
  late Timer _timer;
  bool _isRunning = false;

  @override
  void dispose() {
    _timer.cancel(); // Cancel the timer when the state is
disposed
    super.dispose();
  }

  void _startTimer() {
    setState(() {
      _isRunning = true;
    });
    _timer = Timer.periodic(Duration(seconds: 1), (timer) {
      if (_minutes == 0 && _seconds == 0) {
        _resetTimer();
```

```
        return;
      }
      if (_seconds == 0) {
        setState(() {
          _minutes--;
          _seconds = 59;
        });
      } else {
        setState(() {
          _seconds--;
        });
      }
    });
  }

  void _stopTimer() {
    _timer.cancel();
    setState(() {
      _isRunning = false;
    });
  }

  void _resetTimer() {
    _timer.cancel();
    setState(() {
      _isRunning = false;
      _minutes = 25;
      _seconds = 0;
    });
  }

  String _formatTime(int time) {
    return time < 10 ? '0$time' : '$time';
  }

  @override
```

```
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Row(
          children: [
            Image.asset(
              'assets/images/pomodoro_logo.png',
              width: 40,
              height: 40,
            ),
            SizedBox(width: 30),
            Text('Pomodoro Timer'),
          ],
        ),
        actions: [
          IconButton(
            icon: Icon(Icons.settings),
            onPressed: () {
              // Add settings action
            },
          ),
        ],
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            SizedBox(height: 20),
            Text(

'${_formatTime(_minutes)}:${_formatTime(_seconds)}',
              style: TextStyle(fontFamily: 'Roboto',
                  fontWeight: FontWeight.bold,
                  fontSize: 60),
            ),
            SizedBox(height: 20),
```

```
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: <Widget>[
                ElevatedButton(
                  onPressed: _isRunning ? null : _startTimer,
                  child: Text('Start'),
                ),
                ElevatedButton(
                  onPressed: _isRunning ? _stopTimer : null,
                  child: Text('Stop'),
                ),
                ElevatedButton(
                  onPressed: _resetTimer,
                  child: Text('Reset'),
                ),
              ],
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) =>
LoginPage()),
                );
              },
              child: Text('Login'),
            ),
            SizedBox(height: 20),
            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) =>
SignUpPage()),
                );
```

```
              },
              child: Text('Sign Up'),
            ),
          ],
        ),
      ),
    );
  }
}

class LoginPage extends StatelessWidget {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Login'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextFormField(
                controller: _emailController,
                decoration: InputDecoration(labelText: 'Email'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your email';
```

```
                    }
                    return null;
                  },
                ),
                TextFormField(
                  controller: _passwordController,
                  obscureText: true,
                  decoration: InputDecoration(labelText:
'Password'),
                  validator: (value) {
                    if (value!.isEmpty) {
                      return 'Please enter your password';
                    }
                    return null;
                  },
                ),
                SizedBox(height: 20),
                ElevatedButton(
                  onPressed: () {
                    if (_formKey.currentState!.validate()) {
                      // Perform login action
                      String email = _emailController.text;
                      String password = _passwordController.text;
                      // You can add your login logic here
                      // For now, navigate to the timer page
                      Navigator.pushReplacement(
                        context,
                        MaterialPageRoute(builder: (context) =>
PomodoroTimer()),
                      );
                    }
                  },
                  child: Text('Login'),
                ),
                SizedBox(height: 20),
                TextButton(
```

```
                onPressed: () {
                  Navigator.pop(context);
                },
                child: Text('Back to Timer'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

class SignUpPage extends StatelessWidget {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Sign Up'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Form(
          key: _formKey,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextFormField(
                controller: _emailController,
                decoration: InputDecoration(labelText: 'Email'),
```

```
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your email';
                  }
                  return null;
                },
              ),
              TextFormField(
                controller: _passwordController,
                obscureText: true,
                decoration: InputDecoration(labelText:
'Password'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your password';
                  }
                  return null;
                },
              ),
              SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  if (_formKey.currentState!.validate()) {
                    // Perform sign-up action
                    String email = _emailController.text;
                    String password = _passwordController.text;
                    // You can add your sign-up logic here
                    // For now, navigate to the login page
                    Navigator.pushReplacement(
                      context,
                      MaterialPageRoute(builder: (context) =>
LoginPage()),
                    );
                  }
                },
                child: Text('Sign Up'),
```

```
              ),
              SizedBox(height: 20),
              TextButton(
                onPressed: () {
                  Navigator.pop(context);
                },
                child: Text('Back to Timer'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```


Pomodoro Timer
localhost:53335
Sign Up
Email
2021.raghvendra.tripathi@ves.ac.in
Password
Sign Up
Back to Timer

Conclusion: Thereby we have made a form which actually is a login and sign up form. Also the form validation is available.