EXPERIMENT NO.4

Experiment No 4 **To create an interactive Form using Form widget**	
ROLL NO	
NAME	
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPE D	

2

Aim: To create an interactive Form using Form widget

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.

Username Validator:
The username validator ensures that the username field is not left empty. This is crucial for creating a unique identifier for each user. Users are prompted to enter a username, and if the field is left blank, a validation error is displayed, prompting the user to fill in the required information.

Email Validator:
The email validator ensures that the email entered by the user follows a valid email format. It checks if the email field is empty and then uses a regular expression pattern to validate the email format. This validation ensures that users provide a properly formatted email address, helping maintain communication integrity within the application.

Password Validator:
The password validator ensures that the password provided by the user meets certain criteria, such as a minimum length requirement. In this case, the validator checks if the password field is empty and whether the length of the password is at least 8 characters long. This validation helps enhance security by ensuring that users create strong passwords to protect their accounts.

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.

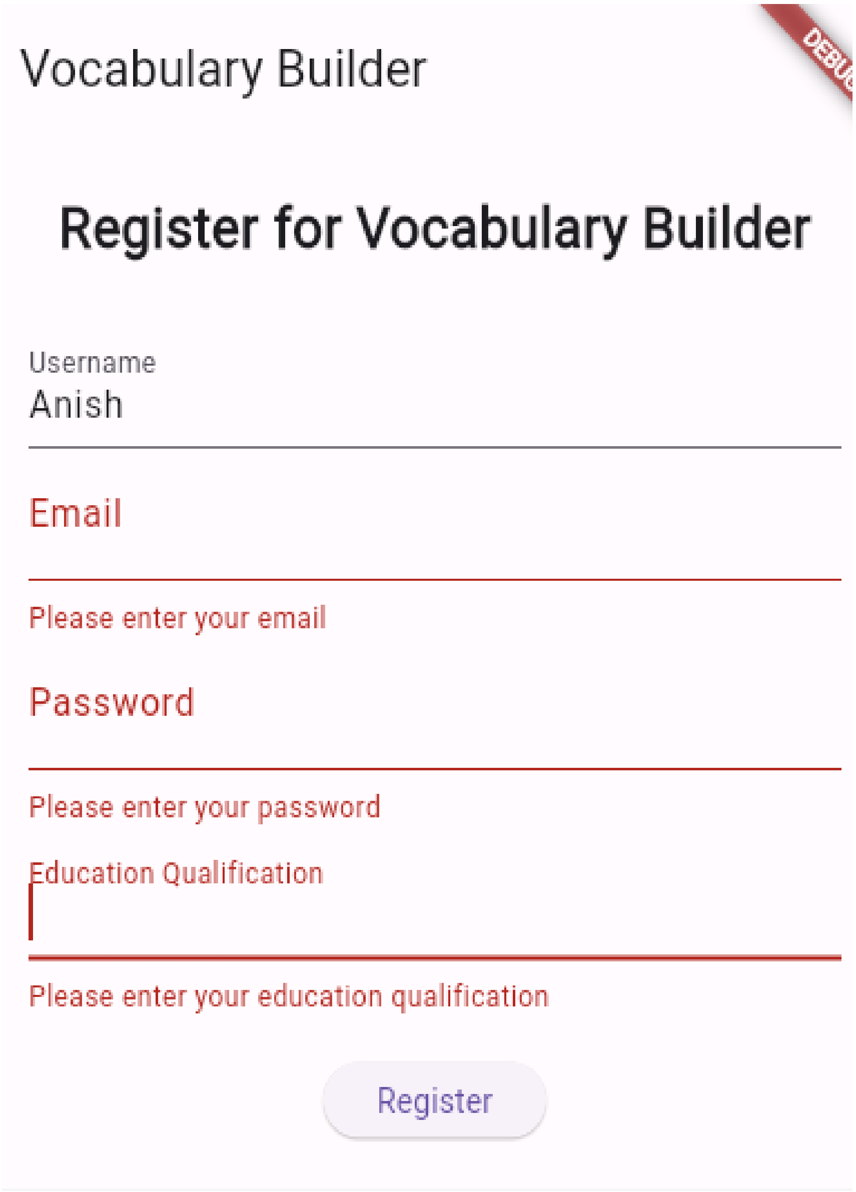
Submit Button:
The Submit button in a Flutter form is typically implemented using a button widget, such as ElevatedButton or TextButton. When pressed, the Submit button triggers the form submission process. Before submitting the form, the form's state is checked using the GlobalKey associated with the 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(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Vocab Builder',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: RegisterPage(),
    );
  }
}

class RegisterPage extends StatefulWidget {
  @override
  _RegisterPageState createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  final _formKey = GlobalKey<FormState>();

  String _username = "";
  String _email = "";
  String _password = "";
  String _educationQualification = "";

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Vocabulary Builder'),
      ),
      body: Form(
        key: _formKey,
        autovalidateMode: AutovalidateMode.onUserInteraction,
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Center(
                child: Text(
                  'Register for Vocabulary Builder',
                  style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
                ),
              ),
              SizedBox(height: 20.0),
              TextFormField(
                decoration: InputDecoration(labelText: 'Username'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your username';
                  }
                  return null;
                },
                onSaved: (value) => _username = value!,
              ),
              TextFormField(
                keyboardType: TextInputType.emailAddress,
                decoration: InputDecoration(labelText: 'Email'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your email';
                  } else if
(!RegExp(r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+\.[a-zA-Z]+").hasMatch(val
ue!)) {
                    return 'Please enter a valid email address';
                  }
                  return null;
                },
                onSaved: (value) => _email = value!,
              ),
              TextFormField(
                obscureText: true,
                decoration: InputDecoration(labelText: 'Password'),
                validator: (value) {
                  if (value!.isEmpty) {
                    return 'Please enter your password';
                  } else if (value!.length < 8) {
                    return 'Password must be at least 8 characters';
                  }
```

```
    return null;
  },
  onSaved: (value) => _password = value!,
),
TextFormField(
  decoration: InputDecoration(labelText: 'Education Qualification'),
  validator: (value) {
    if (value!.isEmpty) {
      return 'Please enter your education qualification';
    }
    return null;
  },
  onSaved: (value) => _educationQualification = value!,
),
SizedBox(height: 20.0),
ElevatedButton(
  onPressed: () {
    if (_formKey.currentState!.validate()) {
      _formKey.currentState!.save();
      // Handle registration logic here (e.g., send data to server)
      showDialog(
        context: context,
        builder: (BuildContext context) {
          return AlertDialog(
            title: Text('Registration Successful'),
            content: Text('Welcome $_username! Your account has been successfully registered.'),
            actions: [
              TextButton(
                onPressed: () {
                  Navigator.of(context).pop();
                  Navigator.push(
                    context,
                    MaterialPageRoute(builder: (context) => WelcomePage()),
                  );
                },
                child: Text('OK'),
              ),
            ],
          );
        },
      );
    }
  },
  child: Text('Register'),
              ),
            ],
          ),
        ),
      ),
    );
  }
}

class WelcomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Welcome'),
      ),
      body: Center(
        child: Text('Welcome to your Vocabulary Builder!'),
      ),
    );
  }
}
```

Output:



Vocabulary Builder

Register for Vocabulary Builder

Username
Anish

Email

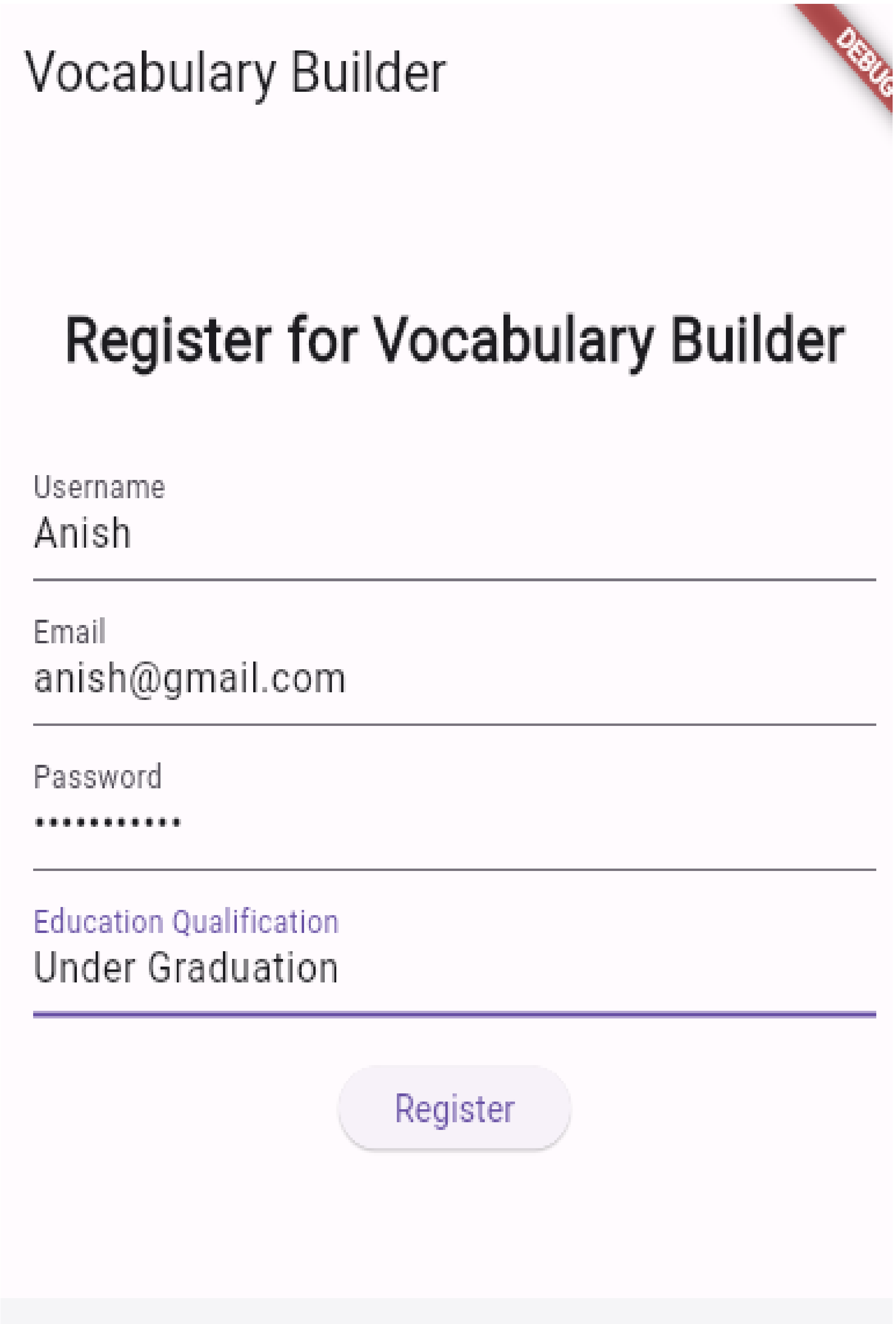
Please enter your email

Password

Please enter your password

Education Qualification

Please enter your education qualification



Vocabulary Builder

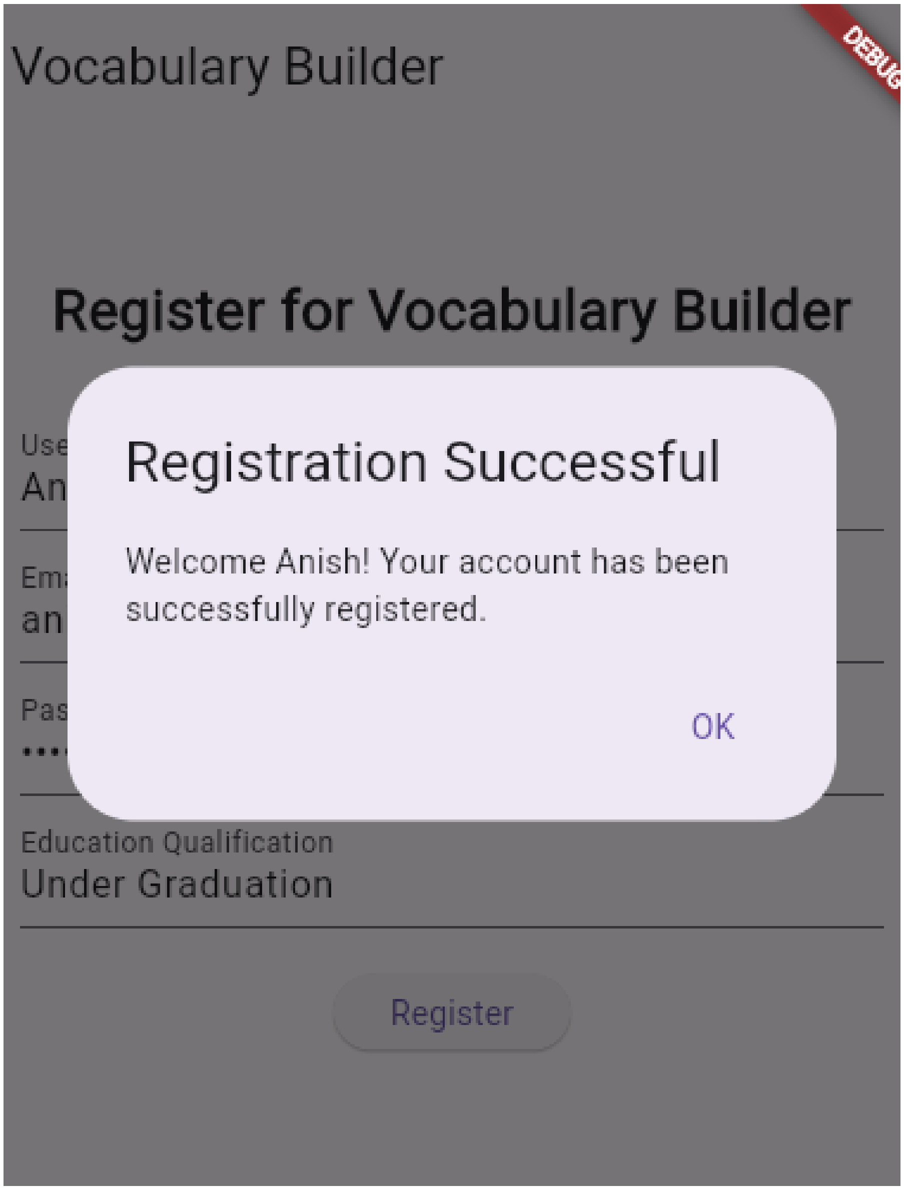
Register for Vocabulary Builder

Username
Anish

Email
anish@gmail.com

Password
•••••••••••

Education Qualification
Under Graduation

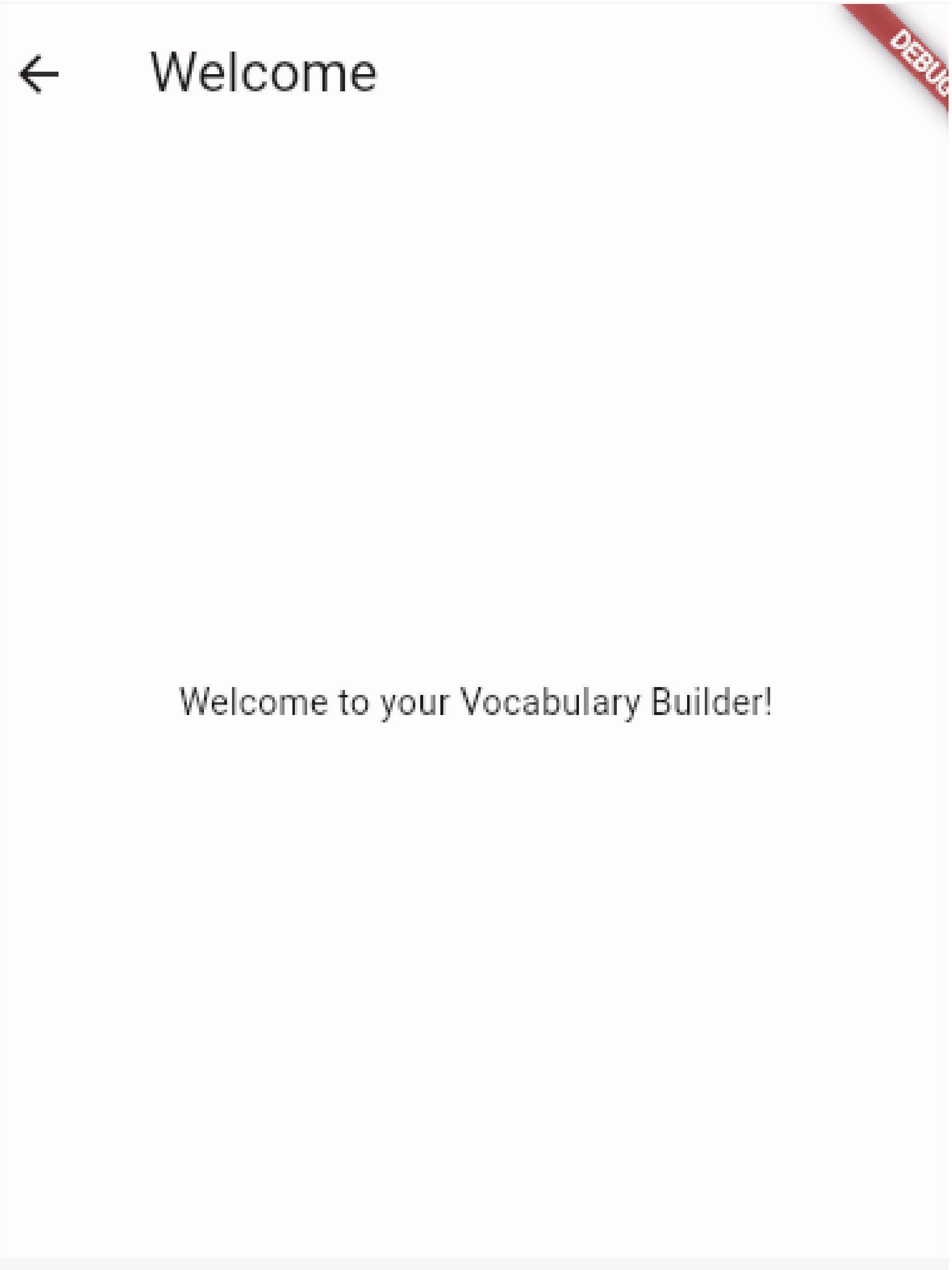


Register for Vocabulary Builder

Registration Successful

Welcome Anish! Your account has been successfully registered.

Education Qualification
Under Graduation



Welcome

Welcome to your Vocabulary Builder!

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.