

## Experiment No: 4

**Name:-** Vishal Gori

**Roll No.:** 18

**Batch:** A

**Division:-** D15B

**AIM:-** To create interactive form using Flutter Widgets

### **THEORY:-**

#### **Form Widget:**

- Groups and manages form fields like text, checkboxes, etc.
- Handles validation, saving, and resetting.
- Use a GlobalKey for advanced control.

#### **Form Fields:**

- Individual input controls like text fields, menus, etc.
- Each holds its own value and behavior.
- Popular options: TextFormField, DropdownButtonFormField, CheckboxFormField, RadioGroupFormField.

#### **Validation:**

- Ensure user input is correct.
- Define validator function for each field or trigger form-level validation.
- Display error messages clearly.

#### **Additional Points:**

- Customize appearance with themes and decorations.
- Manage focus with FocusNode.
- Create custom fields for specific needs.

### **CODE:-**

**main.dart**

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

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

```
}
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: MyHomePage(),  
      theme: ThemeData(  
        primaryColor: Colors.blue,  
        hintColor: Colors.blueAccent,  
        inputDecorationTheme: InputDecorationTheme(  
          border: OutlineInputBorder(),  
          contentPadding:  
            EdgeInsets.symmetric(vertical: 12.0, horizontal: 16.0),  
        ),  
      ),  
    );  
  }  
}
```

```
class MyHomePage extends StatefulWidget {  
  @override  
  _MyHomePageState createState() => _MyHomePageState();  
}
```

```
class _MyHomePageState extends State<MyHomePage> {  
  final _formKey = GlobalKey<FormState>();  
  String _name = "";  
  String _email = "";  
  String _password = "";  
  String _confirmPassword = "";  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text('Basic Form Validation'),  
      ),  
      body: SingleChildScrollView(  
        padding: const EdgeInsets.all(16.0),  
      ),  
    );  
  }  
}
```

```

child: Form(
  key: _formKey,
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      TextFormField(
        decoration: InputDecoration(
          labelText: "Name",
        ),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return "Please enter your name";
          }
          if (RegExp(r'\d').hasMatch(value)) {
            return "Name should not contain numbers";
          }
          return null;
        },
        onSave: (value) => _name = value!,
      ),
      SizedBox(height: 16.0),
      TextFormField(
        decoration: InputDecoration(
          labelText: "Email",
        ),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return "Please enter your email";
          }
          if (!RegExp(
            r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*/+=?^_`{|}~-]+@[a-zA-Z0-9-]+\.[a-zA-Z]+"
          ).hasMatch(value!)) {
            return "Enter valid Email";
          }
          return null;
        },
        onSave: (value) => _email = value!,
      ),
      SizedBox(height: 16.0),
    ],
  ),
),

```

```

TextFormField(
  decoration: InputDecoration(
    labelText: "Password",
  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return "Please enter your password";
    }
    if (value.length < 8) {
      return "Password length should be more than 8";
    }
    return null;
  },
  obscureText: true,
  onSave: (value) => _password = value!,
),
 SizedBox(height: 16.0),
 TextFormField(
  decoration: InputDecoration(
    labelText: "Confirm Password",
  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return "Please confirm your password";
    }
    if (value != _password) {
      return "Passwords do not match";
    }
    return null;
  },
  obscureText: true,
  onSave: (value) => _confirmPassword = value!,
),
 SizedBox(height: 32.0),
 ElevatedButton(
  onPressed: () {
    final isValid = _formKey.currentState!.validate();
    if (isValid) {
      _formKey.currentState!.save();
    }
  },
)

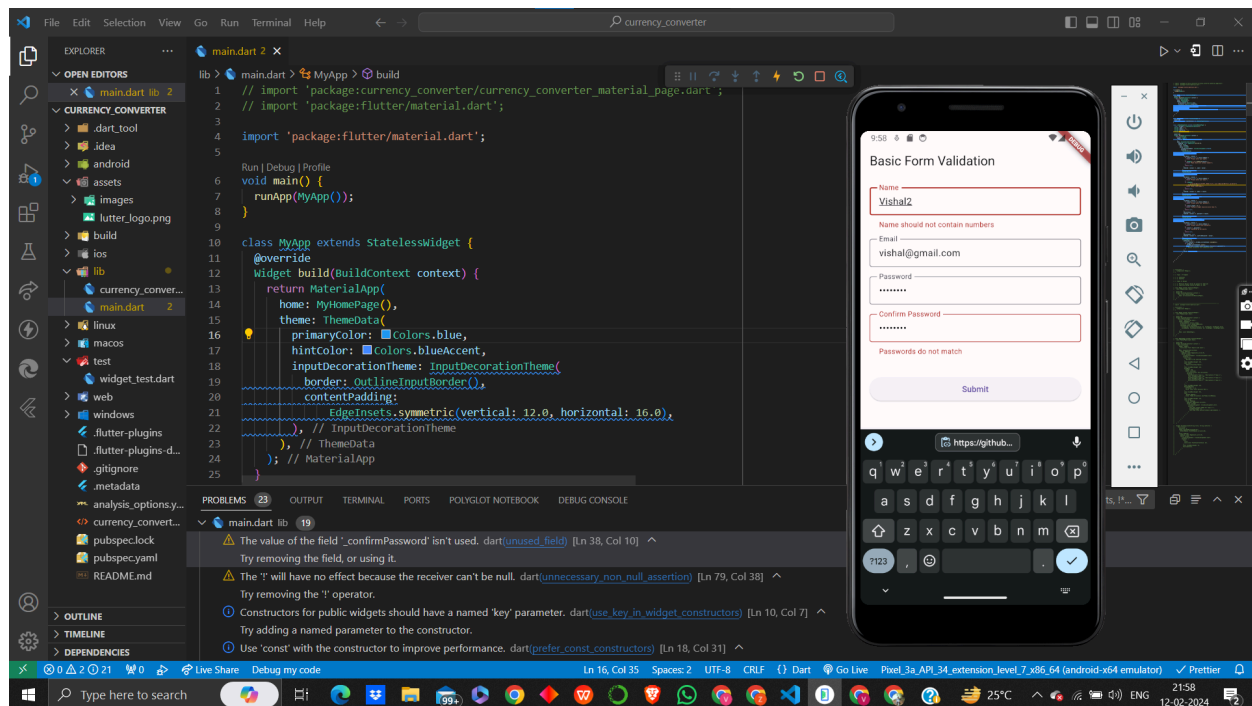
```

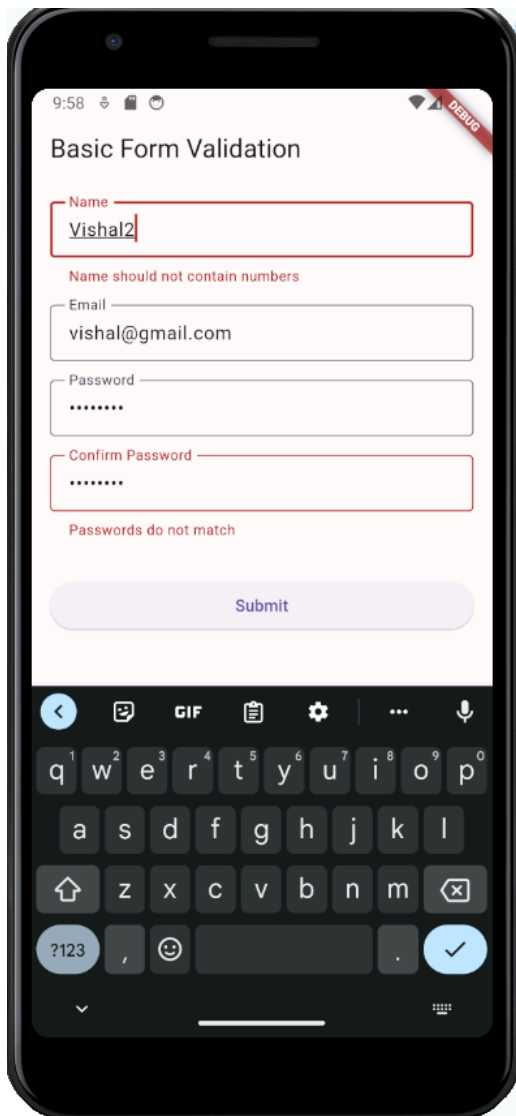
```

        print("Name: $_name, Email: $_email, Password:
$_password");

        // Handle successful form submission here
    }
},
child: Text('Submit'),
),
),
),
),
);
}
}

```





## CONCLUSION:-

In conclusion, the Flutter form code provided implements basic form validation with TextFormField widgets encapsulated in a Scaffold and MaterialApp structure. The design ensures clear user input with validation messages and facilitates form submission with concise error handling. By leveraging Flutter's built-in form handling mechanisms, the code delivers a straightforward and user-friendly experience for data input and validation.