WEEK-7

a) Design a form with various input fields.

AIM: To design a form with various input fields.

DESCRIPTION:

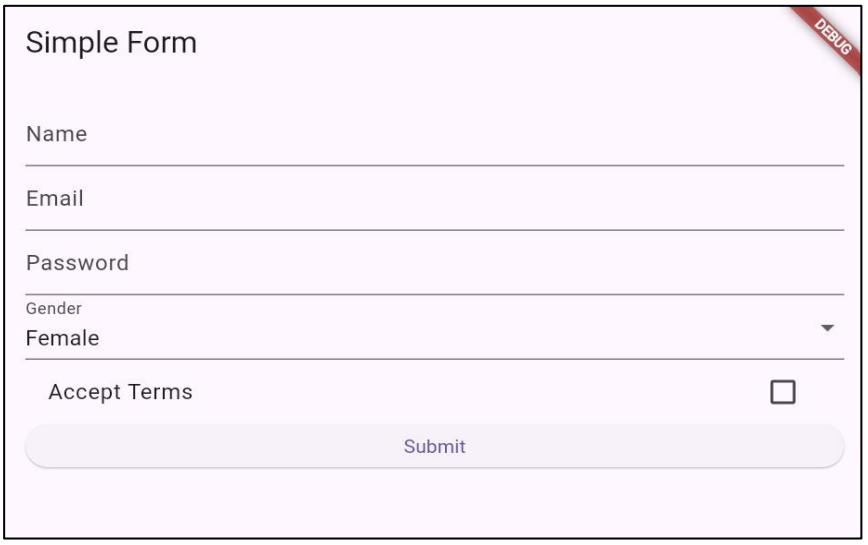
This form is designed to collect user information in a structured and user-friendly way. It includes input fields such as full name, email, phone number, and date of birth. Address details like street, city, state, zip code, and country are also included. For account creation, fields for username, password, and confirmation are provided. Users can select their gender, known languages, and preferred contact method through radio buttons and checkboxes. File upload options are available for profile pictures and documents like resumes. A text area allows for additional comments, and users must agree to terms via a checkbox. The form ends with submit and reset buttons, with validation and accessibility features built in for a smooth experience.

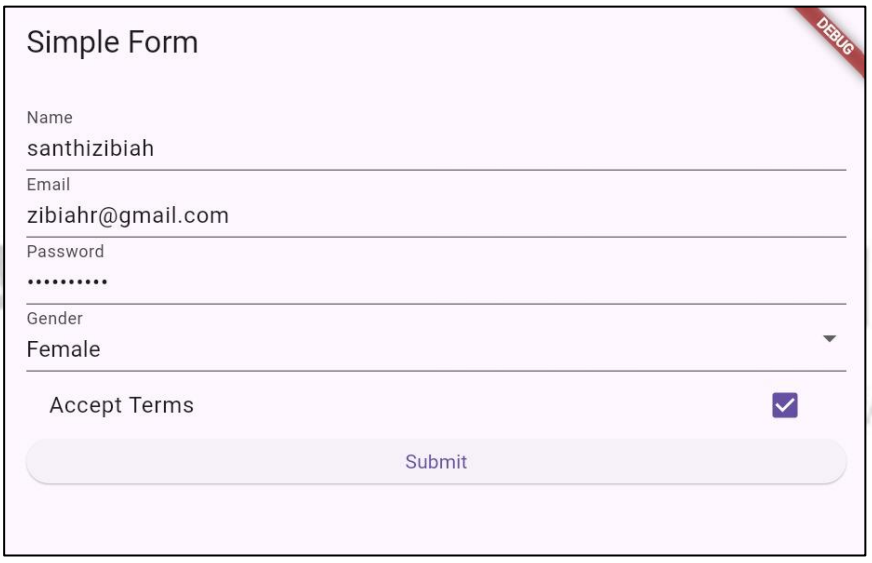
SOURCECODE:

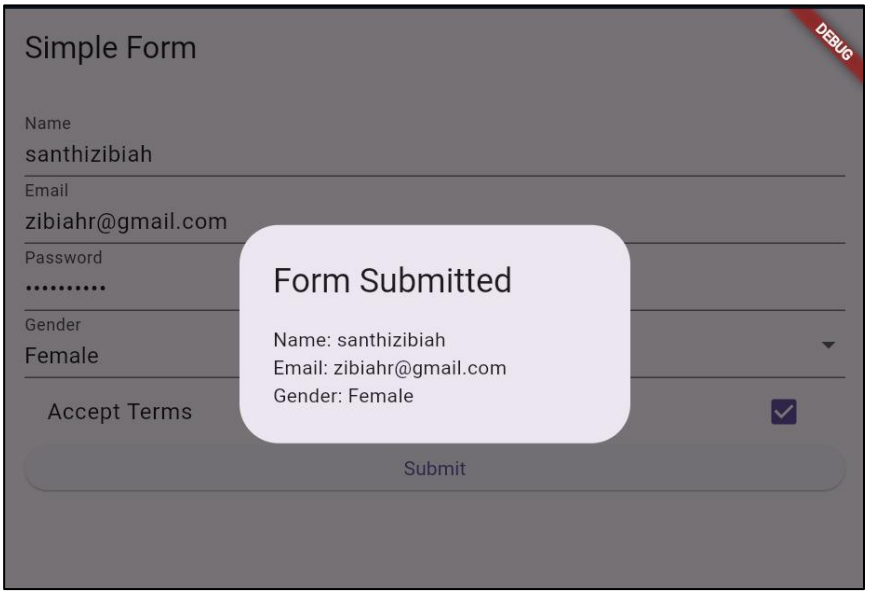
```
import 'package:flutter/material.dart';

void main() => runApp(MaterialApp(home: SimpleForm()));

class SimpleForm extends StatefulWidget {
  @override
  _SimpleFormState createState() => _SimpleFormState();
}

class _SimpleFormState extends State<SimpleForm> {
  final _formKey = GlobalKey<FormState>();
  String name = '';
  String email = '';
  String password = '';
  String gender = 'Male';
  bool accepted = false;
  final genders = ['Male', 'Female', 'Other'];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text('Simple Form')),
      body: Padding(
        padding: EdgeInsets.all(16),
        child: Form(
          key: _formKey,
          child: ListView(
            children: [
              TextFormField(
                decoration: InputDecoration(labelText: 'Name'),
                onSaved: (val) => name = val ?? '',
                validator: (val) => val!.isEmpty ? 'Enter name' : null,
              ),
              TextFormField(
                decoration: InputDecoration(labelText: 'Email'),
                keyboardType: TextInputType.emailAddress,
```

```
            onSaved: (val) => email = val ?? '',
            validator: (val) => val!.contains('@') ? null : 'Enter valid email',
          ),
          TextFormField(
            decoration: InputDecoration(labelText: 'Password'),
            obscureText: true,
            onSaved: (val) => password = val ?? '',
            validator: (val) => val!.length < 6 ? 'Min 6 characters' : null,
          ),
          DropdownButtonFormField<String>(
            initialValue: gender, // Changed from value to initialValue
            items: genders
                .map<DropdownMenuItem<String>>(
                  (g) => DropdownMenuItem<String>(value: g, child: Text(g)))
                .toList(),
            onChanged: (String? val) => setState(() => gender = val ?? 'Male'),
            decoration: InputDecoration(labelText: 'Gender'),
          ),
          CheckboxListTile(
            title: Text('Accept Terms'),
            value: accepted,
            onChanged: (bool? val) => setState(() => accepted = val ?? false),
          ),
          ElevatedButton(
            child: Text('Submit'),
            onPressed: () {
              if (_formKey.currentState!.validate() && accepted) {
                _formKey.currentState!.save();
                showDialog(
                    context: context,
                    builder: (_) => AlertDialog(
                          title: Text('Form Submitted'),
                          content:
                              Text('Name: $name\nEmail: $email\nGender: $gender'),
                        ));
              } else if (!accepted) {
                ScaffoldMessenger.of(context).showSnackBar(
                  SnackBar(content: Text('Please accept terms')),
                );
              }
            },
          )
        ],
      ),
    ),
  ),
 );
 }
}
```

OUTPUT:


Simple Form
DEBUG
Name
Email
Password
Gender
Female
Accept Terms
Submit


Simple Form
DEBUG
Name
santhizibiah
Email
zibiahr@gmail.com
Password
Gender
Female
Accept Terms
Submit


Simple Form
DEBUG
Name
santhizibiah
Email
zibiahr@gmail.com
Password
Gender
Female
Accept Terms
Submit
Form Submitted
Name: santhizibiah
Email: zibiahr@gmail.com
Gender: Female

b) Implement input validation and error messages using Form and validator.

AIM: To implement input validation and error messages using Form and validator.

DESCRIPTION: Input validation is added to the form using validators that check if the user's entries meet specific rules, such as correct email format, required fields not being empty, and password length. When a field fails validation, an error message appears to guide the user to fix the mistake, like "Invalid email" or "This field is required." This helps ensure the data entered is accurate and complete before submission.

SOURCECODE:

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

void main() => runApp(MaterialApp(home:ValidationForm()));
class ValidationForm extends StatefulWidget {
  @override
  _ValidationFormState createState() => _ValidationFormState();
}
class _ValidationFormState extends State<ValidationForm> {
  final _formKey = GlobalKey<FormState>();
  String email='';
  String password='';

  @override
  Widget build(BuildContext context) {
    return Scaffold(
    appBar:AppBar(title:Text('Validation Form')),
    body:Padding(
    padding:EdgeInsets.all(16),
    child: Form(
    key: _formKey,
    child: Column(
    children:[
      //Email input with validator
      TextFormField(
      decoration: InputDecoration(labelText:'Email'),
      keyboardType: TextInputType.emailAddress,
      validator:(val){
        if(val == null || val.isEmpty) return 'Email is required..';
        if(!val.contains('@')) return 'Enter a valid Email..';
        return null;
      },
      onSaved: (val) => email=val??''
      ),
      SizedBox(height: 16),
      //Password input with validator
      TextFormField(
      decoration: InputDecoration(labelText:'Password'),
      obscureText: true,
      validator:(val) {
        if(val == null || val.isEmpty) return 'Password is required..';
        if(val.length<6) return 'Password must be atleast 6 characters';
        return null;
```

```
    },
    onSaved:(val) => password=val??",
    ),
    SizedBox(height:24),
    ElevatedButton(
    child: Text('Submit'),
    onPressed:(){
      if(_formKey.currentState!.validate()){
        //If valid, save the form fileds
        _formKey.currentState!.save();
        showDialog(
        context:context,
        builder:(_) => AlertDialog(
        title: Text('Success'),
        content: Text('Email:$email\nPassword:$password'),));
      }
    })
  ])));
 }
}
```

OUTPUT: