

Experiment no. :- 04

Aim :- To create an interactive Form using form widget

Theory :-

Form Widgets:

Form widgets are essential components of interactive forms, offering a range of input elements such as text fields, checkboxes, radio buttons, dropdown menus, and more. These widgets empower developers to design forms that cater to specific data input requirements. The flexibility of form widgets allows for the creation of dynamic and user-friendly interfaces, ensuring that the form adapts to the user's needs.

Form Inputs:

Text Fields:

Purpose: Allow users to input general text information.

Attributes: May include specifications such as maximum length, placeholder text, and input type (e.g., email, password).

Checkboxes:

Purpose: Enable users to make multiple selections from a list of options.

Attributes: Each checkbox typically represents a distinct option, and users can choose multiple checkboxes simultaneously .

Radio Buttons:

Purpose: Provide users with exclusive choices within a group.

Attributes: Users can select only one option from the group, making radio buttons suitable for mutually exclusive selections.

Dropdown Menus:

Purpose: Offer a space-efficient way to present a list of options for selection.

Attributes: Users click on a dropdown menu to reveal a list of choices, selecting one option from the list.

Textareas:

Purpose: Allow users to input multiline text, suitable for longer responses or comments

Attributes: Can include settings for the number of rows and columns to determine the size of the textarea.

Date Pickers:

Purpose: Facilitate the selection of dates.

Attributes: Users can choose a specific date from a calendar interface, helping to ensure accurate date input.

File Upload:

Purpose: Enable users to submit files (e.g., images, documents).

Attributes: May include file type restrictions, maximum file size, and a browse button for users to locate and upload files from their device.

Code :-

```
import 'dart:convert';

import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:http/http.dart' as http;

import '../screens/light_mode/home_screen.dart';

final formatter = DateFormat.yMd();
var enteredname="";
var entereddate;

class SignUp extends StatefulWidget {
  const SignUp({Key? key}) : super(key: key); // Corrected super call
  @override
  State<StatefulWidget> createState() {
    return _SignUp();
  }
}

class DT{
  DT({required this.date});
  final DateTime date;
  String get formattefDate {
    return formatter.format(date);
  }
}

class _SignUp extends State<SignUp> {
  DateTime? _selectedDate=DateTime.now();
  void _presentDatePicker() async {
    final now = DateTime.now();
    final firstDate = DateTime(now.year - 1, now.month, now.day);
    // final lastDate = DateTime(now.year + 3, now.month, now.day);
    final pickedDate = await showDatePicker(
      context: context,
      initialDate: now,
      firstDate: firstDate,
      lastDate: now);
    if(_selectedDate!=null){
      setState(() {
        _selectedDate = pickedDate??_selectedDate;
      });
    }
  }
  String get formattedDate {
    return formatter.format(DateTime.now());
  }
}

void savedata()async{

  enteredname=nameController.text;
  final url=Uri.https('flutter-prep-5b74d-default-rtdb.firebaseio.com','user-data.json');

  final response= await http.post(url,headers: {
    'Content-type':'application/json'
  },
  body: json.encode({
```

```

        'name': enteredname,
        'dateofbirth': formatter.format(_selectedDate!)
    })
    );
}

final _dateController = TextEditingController();
final nameController=TextEditingController();

@override
Widget build(BuildContext context) {
  DateTime? selectedDate=DateTime.now();
  // var selectedDate = _selectedDate;

  return Scaffold(
    backgroundColor: Theme.of(context).primaryColorLight,
    body: Center(
      child: Padding(
        padding: const EdgeInsets.all(12.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.min,
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            TextField(
              controller: nameController,
              keyboardType: TextInputType.name,
              decoration: InputDecoration(
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.all(Radius.circular(8))),
                iconColor: Colors.black,
                label: Text(
                  "Add Your Name",
                  style: TextStyle(
                    color: Colors.black,
                    fontStyle: FontStyle.italic,
                  ),
                ),
              ),
              expands: false,
            ),
            const SizedBox(height: 10,),
            const SizedBox(height: 10),
            Row(
              children: [
                Expanded(
                  child: TextField(
                    controller: _dateController,
                    decoration: InputDecoration(
                      label: Text('Enter your D.O.B',style: TextStyle(
                        color: Colors.black,
                        fontStyle: FontStyle.italic,
                      ),
                    ),
                  ),
                ),
                Expanded(
                  child: Row(
                    mainAxisAlignment: MainAxisAlignment.end,
                    crossAxisAlignment: CrossAxisAlignment.center,

```

```

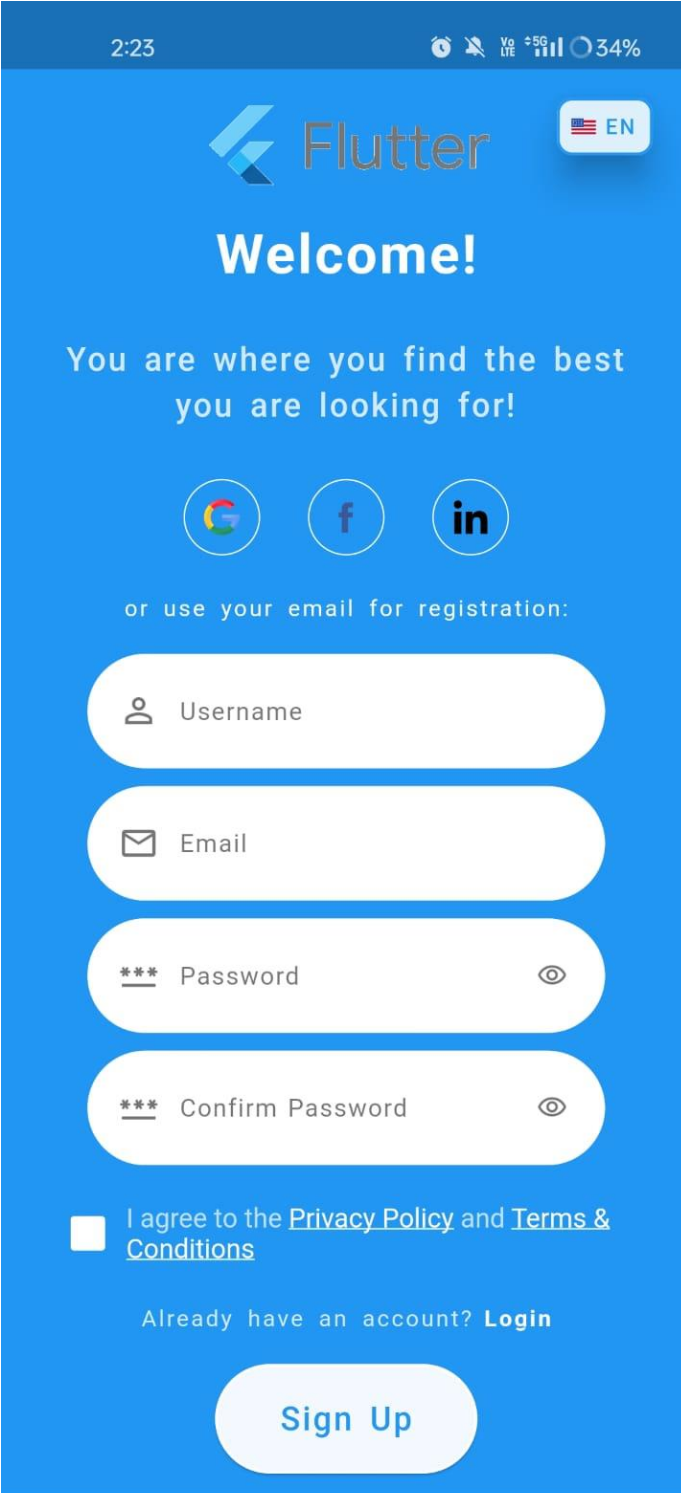
children: [
  Text(

    selectedDate == null
      ? 'No Date Selected'
      : formatter.format(_selectedDate!),
    style: const TextStyle(color: Colors.black)
  ),
  const SizedBox(width: 10),
  IconButton(
    onPressed: () {
      _presentDatePicker();
    },
    icon: const Icon(
      Icons.calendar_month,
      color: Colors.black,
    )),
  // const SizedBox(width: 16),
  // Text(formattedDate),
],
),
),

],
),
const SizedBox(height: 10,),
ElevatedButton(onPressed: (){
  savedata();
  Navigator.of(context).push(MaterialPageRoute(builder: (ctx)=>
    HomeScreen())
  );
}, child: const Text("Next")),
],
),
),
),
);
}
}

```

Output :-

A screenshot of a Flutter application's welcome screen. The background is a solid blue color. At the top, there is a status bar with the time 2:23, various icons (alarm, silent, VoLTE, 5G, signal strength), and a battery level of 34%. Below the status bar, the Flutter logo is on the left, and a language selector button with a US flag and 'EN' is on the right. The word 'Flutter' is in a large, white, sans-serif font. Below the logo, the word 'Welcome!' is written in a large, white, sans-serif font. Underneath, a white text message reads 'You are where you find the best you are looking for!'. There are three circular icons for social media: Google, Facebook, and LinkedIn. Below these, the text 'or use your email for registration:' is displayed. This is followed by four white, rounded rectangular input fields. The first field has a person icon and the label 'Username'. The second field has an envelope icon and the label 'Email'. The third field has three asterisks and the label 'Password', with an eye icon to its right. The fourth field has three asterisks and the label 'Confirm Password', with an eye icon to its right. Below the input fields, there is a checkbox followed by the text 'I agree to the [Privacy Policy](#) and [Terms & Conditions](#)'. At the bottom, the text 'Already have an account? **Login**' is shown. A large, white, rounded rectangular button with the text 'Sign Up' in blue is at the very bottom.