EXPERIMENT NO 4

NAME-Nitish Bhosle

CLASS-D15A

ROLL NO-04

AIM-To create an interactive form using form widget

THEORY-

Forms are essential components of web and mobile applications, allowing users to input and submit data. An interactive form enhances user experience by providing real-time validation, user-friendly input fields, and seamless data handling.

A **Form Widget** is a structured way to manage user input, validate data, and handle submissions efficiently. It provides an interactive interface for users to enter and modify information.

Key Features of Interactive Forms

- User Input Fields: Text fields, dropdowns, checkboxes, radio buttons, and other input elements.
- Real-time Validation: Ensures correct data format before submission.
- Error Handling: Displays messages for invalid inputs.
- Data Submission: Sends user input to a backend or local storage for further processing.
- Dynamic Updates: Auto-fills or adjusts form fields based on user selections.

Components of Form Widget

- Form Container: Wraps all input fields.
- Input Fields: Text fields, number fields, password inputs, email inputs, etc.
- Buttons: Submit and reset buttons to process or clear input.
- Validation Mechanisms: Ensures valid input before submission.

SYNTAX

```
Form(
  key: formKey, // Unique key to manage form state
  child: Column(
    children: [
      TextFormField(
        decoration: InputDecoration(labelText: "Enter your name"),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return "This field cannot be empty";
          }
          return null;
        },
      ),
      SizedBox(height: 10),
      ElevatedButton(
      onPressed: () {
          if (formKey.currentState!.validate()) {
            // Perform form submission action
          }
        },
        child: Text("Submit"),
      ),
    ],
  ),
)
```

Widget Properties

1)key

- Used to uniquely identify the Form widget.
- Typically assigned a GlobalKey<FormState> to manage validation and submissions.

```
Example, final _formKey = GlobalKey<FormState>();


Form(
  key: _formKey,
  child: Column(
    children: [ /* Form fields go here */ ],
  ),
);
```

2)child

- Defines the content inside the Form, usually containing form fields like TextFormField, DropdownButtonFormField, etc.

Example:

```
 Form(

  child: Column(

    children: [

    TextFormField(),

      ElevatedButton(onPressed: () {}, child: Text("Submit")),

    ],

  ),

);
```

3)onchanged

- A callback function that gets triggered when any field inside the form changes.
- Can be used to update state based on form input.

Example:

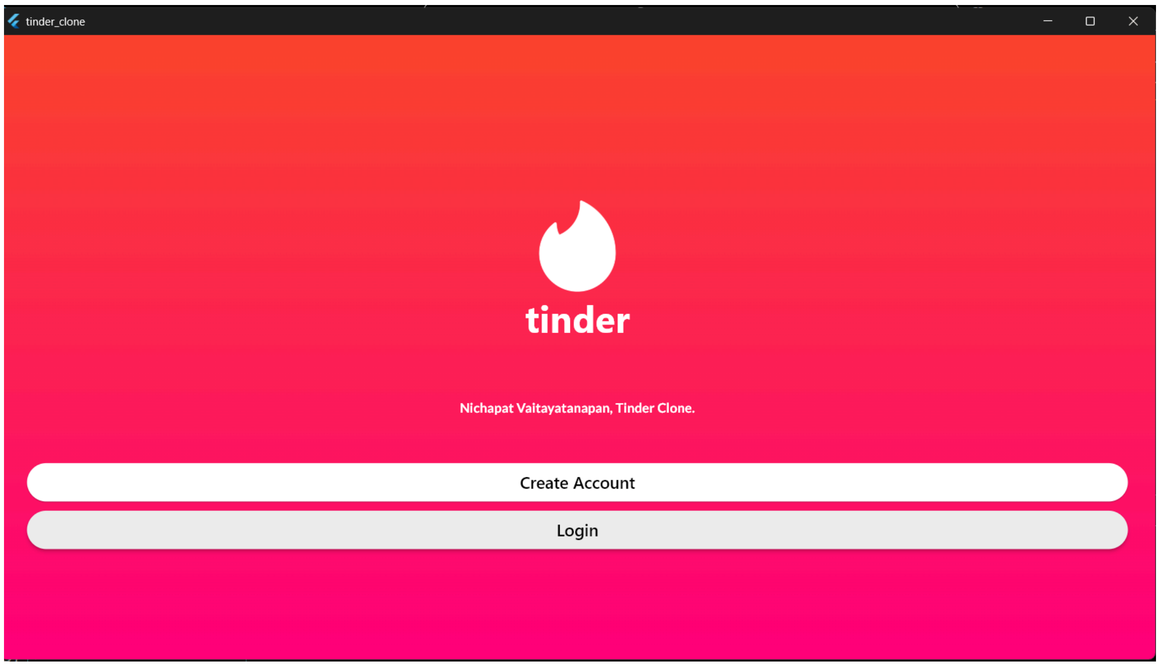
```
Form(
  onChanged: () {
    print("Form data changed!");
  },
  child: TextFormField(),
);
```

CODE

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:tinder_clone/screen/homescreen.dart';

class LoginWithScreen extends StatefulWidget {
  @override
  _LoginWithScreenState createState() => _LoginWithScreenState();
}

class _LoginWithScreenState extends State<LoginWithScreen> {
  final FirebaseAuth _auth = FirebaseAuth.instance;

  Future<UserCredential?> _signInWithGoogle() async {
    try {
      final GoogleSignInAccount? googleUser = await
GoogleSignIn().signIn();
      final GoogleSignInAuthentication googleAuth =
          await googleUser!.authentication;

      final AuthCredential credential =
GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );

      UserCredential userCredential =
          await _auth.signInWithCredential(credential);
      return userCredential;
    } catch (e) {
      print("Google Sign-In Error: $e");
      return null;
    }
  }

  void _handleSignIn() async {
    UserCredential? userCredential = await _signInWithGoogle();
    if (userCredential != null) {
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: ElevatedButton(
          onPressed: _handleSignIn,
          child: Text("Sign in with Google"),
        ),
      ),
    );
  }
}
```

```
import 'package:cloud_firestore/cloud_firestore.dart'
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:tinder_clone/screen/homescreen.dart';

class LoginScreen extends StatelessWidget {
  final FirebaseAuth _auth = FirebaseAuth.instance;
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;

  LoginScreen({super.key}); // Ensure a proper constructor

  Future<void> _signInWithGoogle(BuildContext context) async {
    try {
      final GoogleSignInAccount? googleUser = await
GoogleSignIn().signIn();
      if (googleUser == null) return; // User canceled login

      final GoogleSignInAuthentication googleAuth =
          await googleUser.authentication;

      final AuthCredential credential =
GoogleAuthProvider.credential(
        accessToken: googleAuth.accessToken,
        idToken: googleAuth.idToken,
      );

      UserCredential userCredential =
          await _auth.signInWithCredential(credential);
      User? user = userCredential.user;

      if (user != null) {
        await _firestore.collection('users').doc(user.uid).set({
          'uid': user.uid,
          'name': user.displayName ?? 'No Name',
          'email': user.email ?? 'No Email',
          'profilePic': user.photoURL ?? '',
          'age': 20, // Default age, you can update it later
          'bio': 'Add your bio here!',
          'createdAt': FieldValue.serverTimestamp(),
        }, SetOptions(merge: true));
      }

      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    } catch (e) {
      print("Google Sign-In Error: $e");
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: ElevatedButton(
          onPressed: () => _signInWithGoogle(context),
          child: Text("Sign in with Google"),
        ),
      ),
    );
  }
}
```

OUTPUT


tinder_clone
tinder
Nichapat Vaitayatanapan, Tinder Clone.
Create Account
Login