Aim - To create an interactive form using form widget.

Theory -

Forms are an integral part of all modern mobile and web applications. It is mainly used to interact with the app as well as gather information from the users. They can perform many tasks, which depend on the nature of your business requirements and logic, such as authentication of the user, adding user, searching etc. A form contains text fields, buttons, check boxes, radio buttons, etc.

In order to create a form, you will typically use the 'Form' widget as the top-level container and then add child widgets such as 'TextFormField' and 'Checkbox' to create the input fields.

One of the key features of forms in flutter is their ability to validate user input. The 'TextFormField' widget, for example, has a 'validator' property that can be used to specify a function that checks the input and returns an error message if the input is not valid. You can also use 'GlobalKey <FormState>' to manage the form state and access the form's state.

The basic syntax for creating a form:

```
Form (
        child: colum(
        children : <widget>(
          TextFormField (
              decoration : InputDecoration (labelText
                : 'Enter your email'),
          ),
        TextFormField (
          decoration : InputDecoration (labelText :
               'Enter your password'),
         ),
         RaisedButton (
            child : Tex ('Submit'),
            onPressed : () {},
          ),
        ],
     ),
   );
```

Code -

1. registration_tab_page.dart

```dart
import 'dart:io';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_storage/firebase_storage.dart' as fStorage;
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:image_picker/image_picker.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:users_app/global/global.dart';

import 'package:users_app/splashScreen/my_splash_screen.dart';
import 'package:users_app/widgets/custom_text_field.dart';
import 'package:users_app/widgets/loading_dialog.dart';

class RegistrationTabPage extends StatefulWidget {
  const RegistrationTabPage({Key? key}) : super(key: key);

  @override
  State<RegistrationTabPage> createState() => _RegistrationTabPageState();
}

class _RegistrationTabPageState extends State<RegistrationTabPage> {
  TextEditingController nameTextEditingController = TextEditingController();
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController = TextEditingController();
  TextEditingController confirmPasswordTextEditingController =
      TextEditingController();
  GlobalKey<FormState> formKey = GlobalKey<FormState>();
  String downloadUrlImage = '';

  XFile? imgXFile;
  final ImagePicker imagePicker = ImagePicker();
```

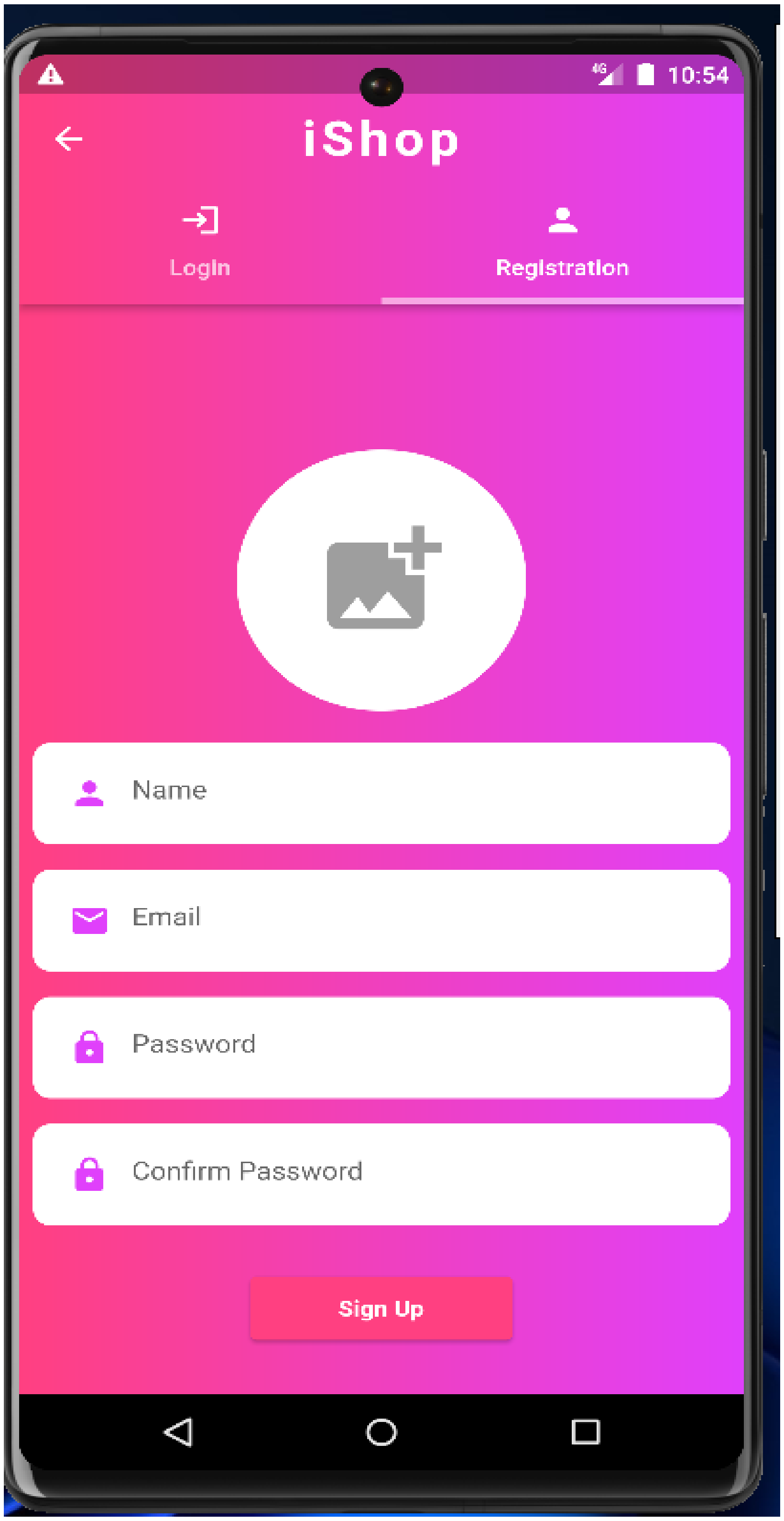
```
getImageFromGallery() async {
  imgXFile = await imagePicker.pickImage(source: ImageSource.gallery);
  setState(() {
    imgXFile;
  });
}

formValidation() async {
  if (imgXFile == null) {
    Fluttertoast.showToast(msg: 'Please select an image.');
  } else
  // image selected
  {
    // password is equal to confirm passwordrd
    if (passwordTextEditingController.text ==
        confirmPasswordTextEditingController.text) {
      // check email, password, confirmpassword and name text fields
      if (nameTextEditingController.text.isNotEmpty &&
          emailTextEditingController.text.isNotEmpty &&
          passwordTextEditingController.text.isNotEmpty &&
          confirmPasswordTextEditingController.text.isNotEmpty) {
        // loading...
        showDialog(
            context: context,
            builder: (c) {
              return LoadingDialogWidget(message: 'Registering your account');
            });

        // upload image to storage
        // za dodjeljivanje imena slici koja se snima koristimo vrijeme (u sString)
        String fileName = DateTime.now().microsecondsSinceEpoch.toString();
        fStorage.Reference storageRef = fStorage.FirebaseStorage.instance
            .ref()
            .child('usersImages')
            .child(fileName);
        fStorage.UploadTask uploadImageTask =
            storageRef.putFile(File(imgXFile!.path));

        fStorage.TaskSnapshot taskSnapshot =
            await uploadImageTask.whenComplete(() {});
        await taskSnapshot.ref.getDownloadURL().then((urlImage) {
          // dodjeljujemo url (string) slike iz storaga
          downloadUrlImage = urlImage;
        });

        // save the user info to firestore database
        saveInformationToDatabase();
        // on complete
        Fluttertoast.showToast(msg: 'Sign Up successfully.');
      } else {
        Navigator.pop(context);
        Fluttertoast.showToast(
            msg: 'Please complete the form. Do not leave text fields empty.');
      }
    } else {
      // password is not equal to confirm passeord
      Fluttertoast.showToast(
          msg: 'Password and Confirm Password are not equal.');
    }
  }
}

saveInformationToDatabase() async {
  //authenticate the user
  User? currentUser;
  await FirebaseAuth.instance
      .createUserWithEmailAndPassword(
          email: emailTextEditingController.text.trim(),
          password: passwordTextEditingController.text.trim())
    .then((auth) {
   currentUser = auth.user;
  }).catchError((errorMessage) {
   Navigator.pop(context);
   Fluttertoast.showToast(msg: 'Error Occurred: \n $errorMessage');
  });

  if (currentUser != null) {
   // save info to database and locally
   saveInfoFirestoreAndLocal(currentUser!);
  }
 }

 saveInfoFirestoreAndLocal(User currentUser) async {
  //to firestore
  FirebaseFirestore.instance.collection('users').doc(currentUser.uid).set({
   'uid': currentUser.uid,
   'email': currentUser.email,
   'name': nameTextEditingController.text.trim(),
   'photoUrl': downloadUrlImage,
   'status': 'approved',
   'userCart': ['initialValue'],
  });
  // locally save with sharedPreferences
  sharedPreferences = await SharedPreferences.getInstance();
  await sharedPreferences!.setString('uid', currentUser.uid);
  await sharedPreferences!.setString('email', currentUser.email!);
  await sharedPreferences!
    .setString('name', nameTextEditingController.text.trim());
  await sharedPreferences!.setString('photoUrl', downloadUrlImage);
  // initial value se nece prikazivati
  await sharedPreferences!.setStringList('userCart', ['initialValue']);

  Navigator.push(
    context, MaterialPageRoute(builder: (c) => MySplashScreen()));
```

```
}

@override
Widget build(BuildContext context) {
 return SingleChildScrollView(
  reverse: true,
  child: Container(
   child: Column(
    children: [
     const SizedBox(height: 12),
     // get capture image

     GestureDetector(
      onTap: () {
       getImageFromGallery();
      },
      child: CircleAvatar(
       radius: MediaQuery.of(context).size.width * 0.20,
       backgroundColor: Colors.white,
       backgroundImage:
         imgXFile == null ? null : FileImage(File(imgXFile!.path)),
       child: imgXFile == null
         ? Icon(
           Icons.add_photo_alternate,
           color: Colors.grey,
           size: MediaQuery.of(context).size.width * 0.20,
          )
         : null,
      ),
     ),
     const SizedBox(height: 12),
     //input field
     Form(
      key: formKey,
      child: Column(
    children: [
      CustomTextField(
        textEditingController: nameTextEditingController,
        iconData: Icons.person,
        hintText: 'Name',
        isObsecure: false,
        enabled: true,
      ),
      CustomTextField(
        textEditingController: emailTextEditingController,
        iconData: Icons.email,
        hintText: 'Email',
        isObsecure: false,
        enabled: true,
      ),
      CustomTextField(
        textEditingController: passwordTextEditingController,
        iconData: Icons.lock,
        hintText: 'Password',
        isObsecure: true,
        enabled: true,
      ),
      CustomTextField(
        textEditingController: confirmPasswordTextEditingController,
        iconData: Icons.lock,
        hintText: 'Confirm Password',
        isObsecure: true,
        enabled: true,
      ),
      const SizedBox(height: 20),
    ],
  ),
),
ElevatedButton(
  style: ElevatedButton.styleFrom(
                backgroundColor: Colors.pinkAccent,
                padding:
                    const EdgeInsets.symmetric(horizontal: 50, vertical: 12),
              ),
              onPressed: () {
                formValidation();
              },
              child: const Text(
                'Sign Up',
                style:
                    TextStyle(color: Colors.white, fontWeight: FontWeight.bold),
              ),
            ),
            const SizedBox(height: 30),
          ],
        ),
      ),
    );
  }
}
```

2. pubspec.yaml

```
name: users_app
description: A new Flutter project.

publish_to: "none"

version: 1.0.0+1

environment:
  sdk: ">=2.18.2 <3.0.0"

dependencies:
  carousel_slider: ^4.1.1
  cart_stepper: ^3.0.2
  cloud_firestore: ^4.0.4
  cupertino_icons: ^1.0.2
  firebase_auth: ^4.1.1
  firebase_core: ^2.1.1
  firebase_messaging: ^14.0.4
  firebase_storage: ^11.0.4
  flutter:
    sdk: flutter
  flutter_staggered_grid_view: ^0.4.1
  fluttertoast: ^8.1.1
  http: ^0.13.5
  image_picker: ^0.8.6
  intl: null
  provider: ^6.0.4
  shared_preferences: ^2.0.15
  smooth_star_rating_nsafe: ^1.0.0+1
dev_dependencies:
  flutter_lints: ^2.0.0
  flutter_test:
    sdk: flutter


flutter:
  uses-material-design: true


  assets:
    - images/
    - slider/
```

Output Screen -


10:54
iShop
Login
Registration
Name
Email
Password
Confirm Password
Sign Up