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

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {


@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
  home: HomePage(),
  theme: ThemeData(
    brightness: Brightness.dark,
  ),
  );
}
}

class HomePage extends StatefulWidget {
@override
_HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
var _formKey = GlobalKey<FormState>();
var isLoading = false;
 late String _email, _age, _country, _hobby;

 String? readEmail ="E-mail";
 String? readAge ="Age";
 String? readCountry ="Country";
 String? readHobby ="Hobby";

 @override
  void initState() {
    // TODO: implement initState
    super.initState();
    read();
  }

  void read() async{
    final prefs = await SharedPreferences.getInstance();
     readEmail = prefs.getString('email');
  readAge = prefs.getString('age');
  readCountry = prefs.getString('country');
  readHobby = prefs.getString('hobby');
  }

void _submit() async{
  final isValid = _formKey.currentState!.validate();
  final prefs = await SharedPreferences.getInstance();

  if (!isValid) {
  return;
  }
  _formKey.currentState!.save();
  await prefs.setString('email', _email);
await prefs.setString('age', _age);
await prefs.setString('country', _country);
await prefs.setString('hobby', _hobby);




  // Obtain shared preferences.
  print(readAge);








}

@override
Widget build(BuildContext context) {
  return Scaffold(
  appBar: AppBar(
    title: Text("Form Validation"),

  ),
  //body
  body: Padding(
    padding: const EdgeInsets.all(16.0),
    //form
    child: Form(
    key: _formKey,
    child: Column(
      children: <Widget>[
      Text(
        "Enter your details",
        style: TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold),
      ),
      //styling
      SizedBox(
        height: MediaQuery.of(context).size.width * 0.1,
      ),
      TextFormField(



        decoration: InputDecoration(labelText: readEmail == "Nothing"?readEmail:readEmail),
        keyboardType: TextInputType.emailAddress,

        onFieldSubmitted: (value) {
        //Validator
        },
        validator: (value) {
        if (value!.isEmpty ||
          !RegExp(r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'*+-/=?^_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+")
            .hasMatch(value)) {
          return 'Enter a valid email!';
        }
        _email = value.toString();
        return null;
        },
      ),
      //box styling
      SizedBox(
        height: 10,
      ),
      //text input
      TextFormField(
        decoration: InputDecoration(labelText: 'Password'),
        keyboardType: TextInputType.visiblePassword,
        onFieldSubmitted: (value) {},
        obscureText: true,
        validator: (value) {
        if (value!.isEmpty) {
          return 'Enter a valid password!';
        }
        return null;
        },
      ),
      SizedBox(
        height: 10,
      ),
      TextFormField(

        decoration: InputDecoration(labelText: readAge == 'Nothing'?readAge:readAge),
        keyboardType: TextInputType.number,
        onFieldSubmitted: (value) {
        //Validator
        },
        validator: (value) {
        if (value!.isEmpty) {
          return 'Text should not be empty';
        }
        _age = value.toString();
        return null;
        },
      ),

      SizedBox(
        height: 10,
      ),

      TextFormField(
        decoration: InputDecoration(labelText: readCountry ==
"Nothing"?readCountry:readCountry),
        keyboardType: TextInputType.text,
        onFieldSubmitted: (value) {
        //Validator
        },
        validator: (value) {
        if (value!.isEmpty) {
          return 'Text should not be empty';
        }
        _country = value.toString();
        return null;
        },
      ),

      SizedBox(
        height: 10,
      ),

      TextFormField(
        decoration: InputDecoration(labelText: readHobby ==
"Nothing"?readHobby:readHobby),
        keyboardType: TextInputType.emailAddress,
        onFieldSubmitted: (value) {
        //Validator
        },
        validator: (value) {
        if (value!.isEmpty) {
          return 'Text should not be empty';
        }
        _hobby = value.toString();
        return null;
        },
      ),
      SizedBox(
        height: 10,
      ),
      ElevatedButton(

        child: Text(
        "Submit",
        style: TextStyle(
          fontSize: 24.0,
        ),
        ),
        onPressed: () => _submit(),
      )
      ],
    ),
    ),
  ),
  );
}
}
```