**Практика 5 - Усовершенствованное приложение для работы со списком дел**

Вам необходимо разработать усовершенствованное приложение для работы со списком дел.

Пример реализации нового проекта представлен на видео ниже:

* На главной странице реализуйте вывод всех записей. Теперь у каждой записи будет название и дополнительный текст. Подобную реализацию можно сделать используя ряд или столбец при выводе текстовых данных в списке.
* Переход на вторую страницу происходит по нажатию иконки в шапке приложения.
* На второй странице должна быть форма, что состоит из двух полей. При нажатии на кнопку добавить новая запись должна помещаться в базу данных Firebase, а также окно с формой должно закрываться.

***Помощь:****при добавлении текстовых полей используйте класс [TextField](https://api.flutter.dev/flutter/material/TextField-class.html" \t "_blank). Он не будет корректно добавляться до тех пор, пока вы не поместите его в объект [Container](https://api.flutter.dev/flutter/widgets/Container-class.html" \t "_blank), в котором будет указана ширина (свойство width).*

**Main.dart:**

import 'package:flutter/material.dart';

// Подключени всех сцен

import 'package:flutter\_todo/pages/add\_item.dart';

import 'package:flutter\_todo/pages/main\_screen.dart';

import 'package:firebase\_core/firebase\_core.dart';

void main() async {

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp();

runApp(MaterialApp(

theme: ThemeData(

primaryColor: Colors.deepOrangeAccent,

),

initialRoute: '/',

routes: {

// Маршрутизация

'/': (context) => MainScreen(),

'/add': (context) => AddItem(),

},

));

}

**Add\_item.dart:**

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

class AddItem extends StatefulWidget {

const AddItem({Key key}) : super(key: key);

@override

\_AddItemState createState() => \_AddItemState();

}

class \_AddItemState extends State<AddItem> {

String \_title, \_text;

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.grey[900],

// Шапка

appBar: AppBar(

title: Text('Добавление записи'),

centerTitle: true,

),

body: Container(

margin: EdgeInsets.only(top: 50),

child: Row(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Column(

children: [

Text('Укажите значения', style: TextStyle(color: Colors.white, fontSize: 30),),

Padding(padding: EdgeInsets.only(top: 10)),

// Для текстовых полей нужно указывать ширину

// по этой причине тут создан контейнер и указана в нем ширина

Container(

width: 200,

child: TextField(

style: TextStyle(color: Colors.white),

decoration: InputDecoration(

border: UnderlineInputBorder(),

labelText: 'Введите название',

labelStyle: new TextStyle(

color: Colors.white,

),

suffixIcon: Icon(Icons.text\_fields),

),

onChanged: (value) {

// При вводе информации сохраняем ее в поле класса

setState(() {

\_title = value;

});

},

),

),

Padding(padding: EdgeInsets.only(top: 10)),

// Аналогичное второе поле

Container(

width: 200,

child: TextField(

style: TextStyle(color: Colors.white),

decoration: InputDecoration(

border: UnderlineInputBorder(),

labelText: 'Введите текст',

labelStyle: new TextStyle(

color: Colors.white,

),

suffixIcon: Icon(Icons.text\_format),

),

onChanged: (value) {

setState(() {

// При вводе информации сохраняем ее в поле класса

\_text = value;

});

},

),

),

Padding(padding: EdgeInsets.only(top: 10)),

// Кнопка добавления

ElevatedButton(onPressed: () {

// При нажатии сохраняем новую запись

// В новой записе указываем название и указываем текст

FirebaseFirestore.instance.collection('items').add({'title': \_title, 'text': \_text});

Navigator.pop(context); // Скрываем окно

}, child: Text('Добавить'))

],

),

],

),

)

);

}

}

**Main\_screen.dart:**

import 'package:flutter/material.dart';

import 'package:cloud\_firestore/cloud\_firestore.dart';

class MainScreen extends StatefulWidget {

const MainScreen({Key key}) : super(key: key);

@override

\_MainScreenState createState() => \_MainScreenState();

}

class \_MainScreenState extends State<MainScreen> {

@override

Widget build(BuildContext context) {

return Scaffold(

backgroundColor: Colors.grey[900],

appBar: AppBar(

title: Text('Список дел'),

centerTitle: true,

actions: [

// Создаем иконку в шапке

IconButton(

icon: Icon(Icons.add),

onPressed: () {

// При нажатии переходим на другую страницу

Navigator.pushNamed(context, '/add');

},

)

],

),

body: StreamBuilder( // Вывод всех записей

stream: FirebaseFirestore.instance.collection('items').snapshots(),

builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {

// Здесь проверка на количество элементов

if(snapshot.data.docs.length == 0) return Center(child: Text('Нет записей', style: TextStyle(color: Colors.white),));

return ListView.builder(

itemCount: snapshot.data.docs.length,

itemBuilder: (BuildContext context, int index) {

return Dismissible(

key: Key(snapshot.data.docs[index].id),

child: Card(

child: ListTile(

title: Row( // Вывод не просто текста, а целого ряда

mainAxisAlignment: MainAxisAlignment.start,

children: [

Column(

children: [

// Вывод названия и полного текста

Text(snapshot.data.docs[index].get('title')),

Text(snapshot.data.docs[index].get('text'), style: TextStyle(color: Colors.grey),),

],

)

],

),

trailing: IconButton(

icon: Icon(

Icons.delete\_sweep,

color: Colors.deepOrangeAccent

),

onPressed: () {

FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).delete();

},

),

),

),

onDismissed: (direction) {

FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).delete();

},

);

}

);

},

),

);

}

}