Name:Hayden Cordeiro
Roll No :05
Batch : D

Experiment 12
Develop an application that makes use of Database

Aim:- To develop an application that makes use of database.

Code:
Firebase_api.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:todotrial/models/todo.dart';

class FirebaseApi {
  static String UserID = "";
  static String UserName = "";
  static String UserPhotoURL = "";
  static bool UisLoggedIn = false;

  static Future<String> createTodo(ToDo todo) async {
    final docToDo = FirebaseFirestore.instance.collection(UserID).doc();
    await docToDo.set(todo.toJson());
    todo.key = docToDo.id;
    updateTodo(todo);
  }

  static updateTodo(ToDo todo) async {
    final docToDo =
        FirebaseFirestore.instance.collection(UserID).doc(todo.key).update({
      "isDone": todo.isDone,
      "key": todo.key,
      "text": todo.text,
    }).then((_) {
      print("success updated!");

      // print(todo.isDone);
      // getTodo();
    });
```

```
  }

  static getTodo(update) async {
    QuerySnapshot querySnapshot =
        await FirebaseFirestore.instance.collection(UserID).get();

    var l = querySnapshot.docs.toList();
    // map((DocumentSnapshot docSnapshot){
    //   return ToDo(docSnapshot.data()['key'], docSnapshot.data()['text'],
docSnapshot.data()['isDone']);
    // });

    List<ToDo> l2 = [];
    for (var i = 0; i < l.length; i++) {
      // print(l[i].data());
      l2.add(
          ToDo(l[i].data()['key'], l[i].data()['text'], l[i].data()['isDone']));
    }
    // print(l2);
    update(l2);
  }
}
```

todo.dart

```
class ToDo {
  String key;
  String text;
  bool isDone;

  ToDo(this.key, this.text, this.isDone);

  Map<String, dynamic> toJson() =>
      {
        'key': key,
        'text': text,
        'isDone': isDone
      };

  ToDo fromJson(Map<String, dynamic> json) {
    return ToDo(json['key'], json['text'], json['isDone']);
```

```
  }
}
```

drawer.dart

```
import 'package:flutter/material.dart';
import 'package:todotrial/widgets/createDrawerBodyItem.dart';
import 'package:todotrial/widgets/createDrawerHeader.dart';
import 'package:todotrial/routes/pageRoute.dart';

class NavigationDrawer extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: <Widget>[
          createDrawerHeader(),
          createDrawerBodyItem(
              icon: Icons.home,
              text: 'Home',
              onTap: () => {
                    Navigator.pushReplacementNamed(context, pageRoutes.home),
                  }),
          createDrawerBodyItem(
              icon: Icons.account_circle,
              text: 'Profile',
              onTap: () => {
                    Navigator.pushReplacementNamed(context, pageRoutes.login),
                  }),
          ListTile(
            title: Text('App version 1.2.0'),
            onTap: () {},
          ),
        ],
      ),
    );
  }
}
```

home.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:todotrial/api/firebase_api.dart';
import 'package:todotrial/models/todo.dart';
import 'package:todotrial/pages/drawer.dart';

class Home extends StatefulWidget {
  @override
  _HomeState createState() => _HomeState();
}

class _HomeState extends State<Home> {
  // final todos = [
  //   ToDo('0', "create Project", true),
  //   ToDo('1', "Add a To Do", true),
  //   ToDo('2', "Complete All Your ToDos", false)
  // ];
  @override
  void initState() {
    super.initState();
    FirebaseApi.getTodo(_update);
  }

  var todos = [];

  // FirebaseApi.getTodo();
  void _update(List<ToDo> count) {
    // print("asdf");
    setState(() => todos = count);
  }

  createAlertDialogue(BuildContext context) {
    TextEditingController controllerText = TextEditingController();
    return showDialog(
        context: context,
        builder: (context) {
          return AlertDialog(
            title: Text("Add ToDo"),
            content: TextField(
```

```
          controller: controllerText,
        ),
        actions: <Widget>[
          MaterialButton(
            onPressed: () {
              ToDo tempToDo =
                  ToDo("", controllerText.text.toString(), false);
              FirebaseApi.createTodo(tempToDo);
              Navigator.of(context).pop();
              setState(() {
                print(todos.length);
                todos.add(tempToDo);
                todos.sort((a, b) {
                  if (b.isDone) {
                    return -1;
                  }
                  return 1;
                });
              });
            },
            elevation: 0.5,
            child: Text("Add"),
          )
        ],
      );
    });
}

GestureDetector createTodo(double width, ToDo temp) {
  return GestureDetector(
      onTap: () {
        setState(() {
          ToDo tempToDo = (todos.elementAt(
              todos.indexWhere((element) => element.key == temp.key)));
          tempToDo.isDone = !tempToDo.isDone;
          FirebaseApi.updateTodo(tempToDo);

          todos.sort((a, b) {
            if (b.isDone) {
              return -1;
```

```
      }
      return 1;
    });
  });
},
child: Center(
   child: Container(
      width: 0.95 * width,
      decoration: BoxDecoration(
         color: Colors.white,
         border: Border.all(
           color: Colors.grey[100],
         ),
         borderRadius: BorderRadius.all(Radius.circular(50))),
      padding: EdgeInsets.symmetric(horizontal: 12, vertical: 15),
      margin: EdgeInsets.only(top: 10),
      child: Row(
       mainAxisAlignment: MainAxisAlignment.spaceBetween,
       children: <Widget>[
         Text(
           temp.text,
           style: TextStyle(
             fontSize: 17,
             color: temp.isDone ? Colors.grey : Colors.black,
             decoration:
                temp.isDone ? TextDecoration.lineThrough : null,
           ),
         ),
         Container(
           height: 25,
           width: 25,
           decoration: BoxDecoration(
              color: temp.isDone ? Colors.green : Colors.white,
              borderRadius: BorderRadius.all(Radius.circular(20)),
              border: Border.all(
                color: Colors.grey[300],
              )),
           child: Icon(
             Icons.done,
             size: 16,
```

```
                    color: Colors.white,
                  ),
                )
              ],
            ))));
}

@override
Widget build(BuildContext context) {
  double width = MediaQuery.of(context).size.width;
  // double height = MediaQuery.of(context).size.height;
  final GlobalKey<ScaffoldState> _scaffoldKey =
      new GlobalKey<ScaffoldState>();
  return SafeArea(
      child: Scaffold(
          key: _scaffoldKey,
          drawer: NavigationDrawer(),
          backgroundColor: Color(0xffF4F4F5),
          appBar: AppBar(
            leading: IconButton(
              icon: Icon(
                Icons.subject_sharp,
                color: Color(0xff53A5D5),
                size: 30.0,
              ),
              onPressed: () {
                _scaffoldKey.currentState.openDrawer();
              },
            ),
            elevation: 0.0,
            centerTitle: true,
            backgroundColor: Colors.white,
            title: Text(
              "All Task",
              style: TextStyle(
                color: Colors.black,
              ),
            ),
          ),
          floatingActionButton: FloatingActionButton(
```

```
            onPressed: () {
              createAlertDialogue(context);
            },
            tooltip: 'Add',
            backgroundColor: Color(0xff53A5D5),
            child: Icon(
              Icons.add,
            ),
          ),
          floatingActionButtonLocation:
              FloatingActionButtonLocation.centerFloat,
          body: ListView(
              children: [
            // createTodo(width,false),
          ]..addAll(todos.map((todo) => createTodo(width, todo))))
          // Container(
          //   height: 70,
          // )

          ));
  }
}
```

login.dart

```
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';
import 'package:todotrial/api/firebase_api.dart';
import 'package:todotrial/pages/drawer.dart';
import 'package:google_sign_in/google_sign_in.dart';
import 'package:todotrial/routes/pageRoute.dart';
import 'package:todotrial/utils/shared_pref.dart';
// import 'package:shared_preferences/shared_preferences.dart';

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  bool _isLoggedIn = false;
```

```
final GlobalKey<ScaffoldState> _scaffoldKey =
new GlobalKey<ScaffoldState>();
void _update(bool count) {
  // print("asdf");
  setState(() => _isLoggedIn = count);
}

@override
void initState() {
  super.initState();
  Shared_Pref.GetIsLogged(_update);
}

GoogleSignIn _googleSignIn = GoogleSignIn(scopes: ['email']);

_login() async {
  try {
    await _googleSignIn.signIn();
    setState(() {
      _isLoggedIn = true;
    });
    FirebaseApi.UserID = _googleSignIn.currentUser.email;
    FirebaseApi.UserName = _googleSignIn.currentUser.displayName;
    FirebaseApi.UserPhotoURL = _googleSignIn.currentUser.photoUrl;
    FirebaseApi.UisLoggedIn = true;

    Shared_Pref.SetUID(
        _googleSignIn.currentUser.email,
        _googleSignIn.currentUser.displayName,
        _googleSignIn.currentUser.photoUrl,
        true);
    // print(FirebaseApi.UserID);
    Navigator.pushReplacementNamed(context, pageRoutes.home);
  } catch (err) {
    print(err);
  }
}

_logout() {
  _googleSignIn.signOut();
```

```
  setState(() {
    _isLoggedIn = false;
  });
  FirebaseApi.UserID = "";
  Shared_Pref.SetUID("", "", "", false);
}

@override
Widget build(BuildContext context) {
  // TODO: implement build
  return Scaffold(
    key: _scaffoldKey,
    drawer: NavigationDrawer(),
    backgroundColor: Color(0xffF4F4F5),
    appBar: AppBar(
      leading: IconButton(
        icon: Icon(
          Icons.subject_sharp,
          color: Color(0xff53A5D5),
          size: 30.0,
        ),
        onPressed: () {
          _scaffoldKey.currentState.openDrawer();
        },
      ),
      elevation: 0.0,
      centerTitle: true,
      backgroundColor: Colors.white,
      title: Text(
        "All Task",
        style: TextStyle(
          color: Colors.black,
        ),
      ),
    ),
    body: Center(
        child: _isLoggedIn
            ? Column(
                mainAxisAlignment: MainAxisAlignment.center,
                children: <Widget>[
```

```
                Image.network(
                  FirebaseApi.UserPhotoURL,
                  height: 50.0,
                  width: 50.0,
                ),
                Text(FirebaseApi.UserName),
                OutlineButton(
                  child: Text("Logout"),
                  onPressed: () {
                    _logout();
                  },
                )
              ],
            )
          : Center(
              child: OutlineButton(
                child: Text("Login with Google"),
                onPressed: () {
                  _login();
                },
              ),
            )),
    );
  }
}
```

Sharedpref.dart

```
import 'package:shared_preferences/shared_preferences.dart';
import 'package:todotrial/api/firebase_api.dart';

class Shared_Pref {
  static SharedPreferences _pref;
  static Future init() async {
    _pref = await SharedPreferences.getInstance();
    FirebaseApi.UserID = _pref.getString('UID') ?? "";

    FirebaseApi.UserName = _pref.getString('UName') ?? "";
    FirebaseApi.UserPhotoURL = _pref.getString('UPhoto') ?? "";
    FirebaseApi.UisLoggedIn = _pref.getBool('isLoggedIn') ?? false;
```

```
  }

  static Future SetUID(
      String username, String uName, String uPhoto, bool log) async {
    await _pref.setString('UID', username);
    await _pref.setString('UName', uName);
    await _pref.setString('UPhoto', uPhoto);
    await _pref.setBool('isLoggedIn', log);
  }

  static Future GetIsLogged(update) {
    update(_pref.getBool('isLoggedIn') ?? false);
  }
}
```

createDrawerBodyItem.dart

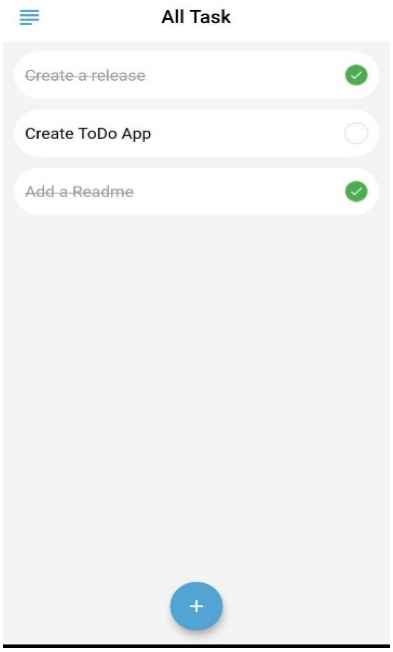
```
import 'package:flutter/material.dart';

Widget createDrawerBodyItem(
    {IconData icon, String text, GestureTapCallback onTap}) {
  return ListTile(
    title: Row(
      children: <Widget>[
        Icon(icon),
        Padding(
          padding: EdgeInsets.only(left: 8.0),
          child: Text(text),
        )
      ],
    ),
    onTap: onTap,
  );
}
```

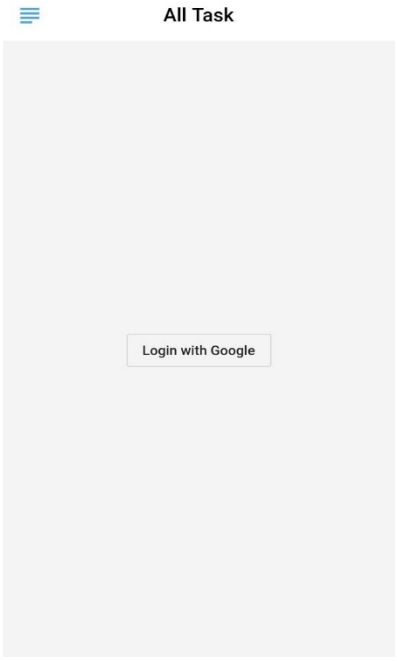
createDrawerHeader.dart

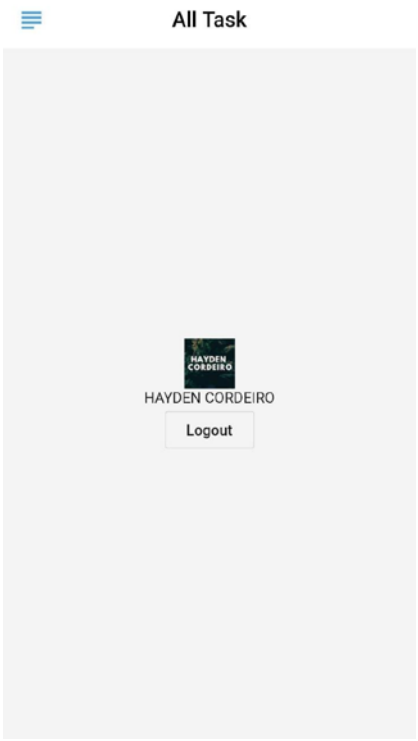
```
import 'package:flutter/material.dart';

Widget createDrawerHeader() {
  return DrawerHeader(
```

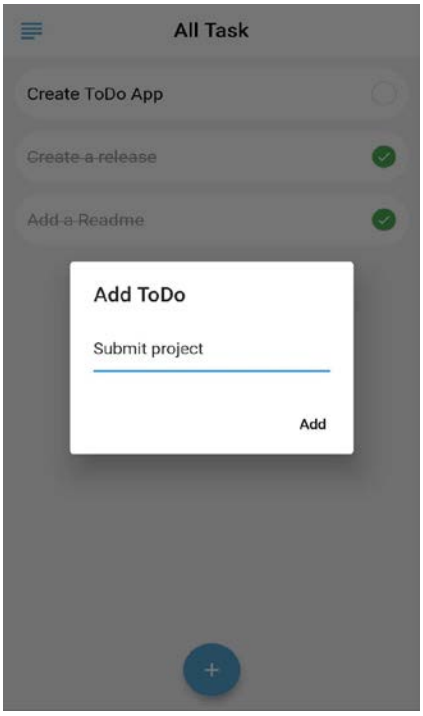
```
    margin: EdgeInsets.zero,
    padding: EdgeInsets.zero,
    decoration: BoxDecoration(
       image: DecorationImage(
          fit: BoxFit.fill,
          image: NetworkImage(
             "https://images.pexels.com/photos/281260/pexels-photo-
281260.jpeg?cs=srgb&dl=pexels-francesco-ungaro-281260.jpg&fm=jpg"))),
    child: Stack(children: <Widget>[
      Positioned(
         bottom: 12.0,
         left: 16.0,
         child: Text("Welcome to ToDo",
            style: TextStyle(
               color: Colors.white,
               fontSize: 20.0,
               fontWeight: FontWeight.w500))),
    ]));
}
```

Output:









Conclusion:

Thus we have implemented a database in Android application using Flutter