**PRACTICAL # 11**

**OBJECT:**

Integrating Firebase database to applications

**THEORY:**

Firebase is a backend service to develop high-quality apps. In this lab, we will integrate the NoSQL database service of Firebase. To use Firebase services, make sure you have installed Google Repository version 26 or higher, with following steps:

Click Tools > SDK Manager.

Click the SDK Tools tab.

Check the Google Repository checkbox, and click OK.

Click OK to install.

Use the Assistant window in Android Studio to add Firebase support, by following steps:

Click Tools > Firebase to open the Assistant window.

Click to expand one of the listed features (Realtime Database), then click the Get Started tutorial to connect to Firebase and add the necessary code to your app.

Steps to add Firebase to an Android app using Firebase Assistant:

1. Open the android studio and click on Tools in the upper left corner.

2. Click on the Firebase option in the drop down menu.

3. A menu will appear on the right side of screen. It will show services that Firebase offers. Choose the desired service.

4. Click Connect to Firebase option in the menu of desired service.

5. Add the dependencies of your service by clicking on the Add [YOUR SERVICE NAME] to the app option.

Figure 1 shows such steps visually.



Figure 1: Integrating Firebase Realtime Database

After adding Firebase Realtime Database dependency, synchronize the project. This may result in errors in the app/build.gradle file. To resolve the errors make sure that a valid version of the Database is added (example: implementation 'com.google.firebase:firebase-database:16.0.1'). With this, you will also need another dependency: implementation 'com.google.firebase:firebase-core:16.0.3'

The app/build.gradle should look like:

*apply plugin: 'com.android.application'*

*apply plugin: 'com.google.gms.google-services'*

*android {*

*compileSdkVersion 28*

*defaultConfig {*

*applicationId "com.example.shan.fbaseapp1"*

*minSdkVersion 23*

*targetSdkVersion 28*

*versionCode 1*

*versionName "1.0"*

*testInstrumentationRunner*

*"android.support.test.runner.AndroidJUnitRunner"*

*}*

*buildTypes {*

*release {*

*minifyEnabled false*

*proguardFiles getDefaultProguardFile('proguardandroid.*

*txt'), 'proguard-rules.pro'*

*}*

*}*

*}*

*dependencies {*

*implementation fileTree(dir: 'libs', include: ['\*.jar'])*

*implementation 'com.android.support:appcompat-v7:28.0.0'*

*implementation 'com.android.support.constraint:constraintlayout:*

*1.1.3'*

*testImplementation 'junit:junit:4.12'*

*androidTestImplementation 'com.android.support.test:runner:1.0.2'*

*androidTestImplementation*

*'com.android.support.test.espresso:espresso-core:3.0.2'*

*implementation 'com.google.firebase:firebase-database:16.0.1'*

*implementation 'com.google.firebase:firebase-core:16.0.3'*

*}*

Before synchronizing the project, you will need to add the googleservices.json configuration file downloaded from Firebase console.

**Integrating Firebase Realtime Database Step by Step:**

**Creating Firebase Project**

Open Firebase Console at https://console.firebase.google.com and create a firebase account to start with. In Firebase console Create a New Project by clicking on the “Create New Project” Button as shown in Figure 2 below.



Figure 2: Creating project on Firebase console

Now you will see Step 1 of 3.

1. In first step, name the project and accept the Firebase terms and click “Continue” button.

2. Second step is Google Analytics for your Firebase project. You can either leave it default enabled or disable it and click Continue. If you disable the Analytics, the third step will not appear and the button will be Create project. In our case, we will just disable the Analytics and create project.

3. If you enabled Analytics, the third step asks your Country and terms and conditions of Analytics. Accept these terms and click Create project as shown in the figure 3.



Figure 3: Configurations of Analytics

Next screen shows your project is ready. Click Continue.

**Linking Android Application with the Firebase Project**

The steps involved to link your project with Firebase are summarized in the figure 4:



Figure 4: Linking your application with Firebase project

On the Project main page, you will be given options to Add an app for the project. Choose

the Android icon as shown in the figure 5.



Figure 5: Choosing platform

You need to register your app with Firebase project.

Step 1- App Details:

The first step, shown in the figure 6 asks for package name. Specify the package name of your app and leave the default fields. Proceed by clicking Register app.



Figure 6: Specifying package name of your android application

Step 2- Copy Configuration File:

In this step, you have a button named Download google-services.json.

Click to download the JSON configuration file. Add the file to your Android App project in Android Studio as shown in figure 7. To do this, you need to change from Android view to Project view.



Figure 7: Download and add JSON configuration file to your Android Application

Step 3- Add Firebase SDK and sync gradle:

The next step is to add Firebase dependencies to gradle and sync it. The instructions are mentioned on the page as shown in figure 8.



Figure 8: Adding Firebase dependencies to gradle

Finally, press "Sync now" in the bar that appears in the IDE. This will take some time. Click next to continue.

Here a possible error may occur: No matching client found for package name 'com.example.shan.firebasedemoapp1'

This is because the package name of the Firebase project did not match the package name of the Android App.

Step 4- Finishing:

On the last step, click Continue to console to finish adding the project.

The code for sample ToDo List application is accessible at:

https://github.com/shanniz/fBaseApp1.git

**ACTIVITIES**

**Activity 1**

Develop a news articles application using Firebase Realtime Database that saves news articles with title and detail on Firebase.

**Activity 2**

Develop computer science educational app that allows to save the computer science MCQs and with their options and correct answers on Firebase Realtime Database.

**REVIEW QUESTIONS**

1. What is Firebase?
2. What are the benefits of Firebase Realtime Database?
3. What is the difference between relational and non-relational databases?
4. What is meant by synchronizing the project?