EXPERIMENT-7[B]

AIM:

Build mobile application serverless database Firebase (cloud-hosted database)

DESCRIPTION:

Firebase is a mobile platform that helps you quickly develop high-quality apps, grow your user base, and earn more money. Firebase consists of complementary features that you can mix and match to fit your needs, with Google Analytics for Firebase at the core. You can explore and integrate Firebase services in your app directly from Android Studio using the **Assistant** window shown in figure 1.

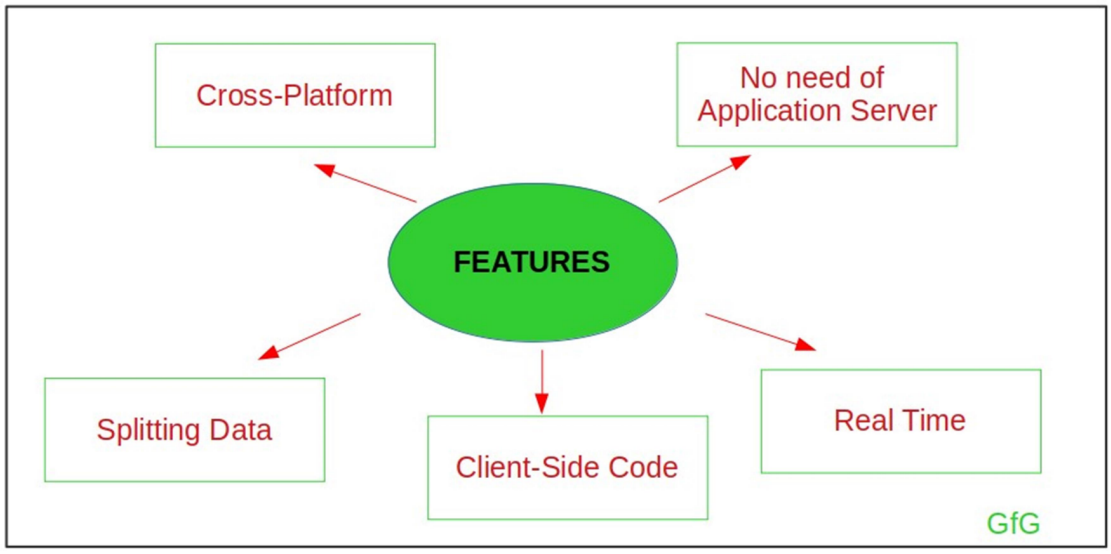
First, make sure you have added Google's Maven repository to your project configuration.

You can open and use the **Assistant** window in Android Studio by following these steps:

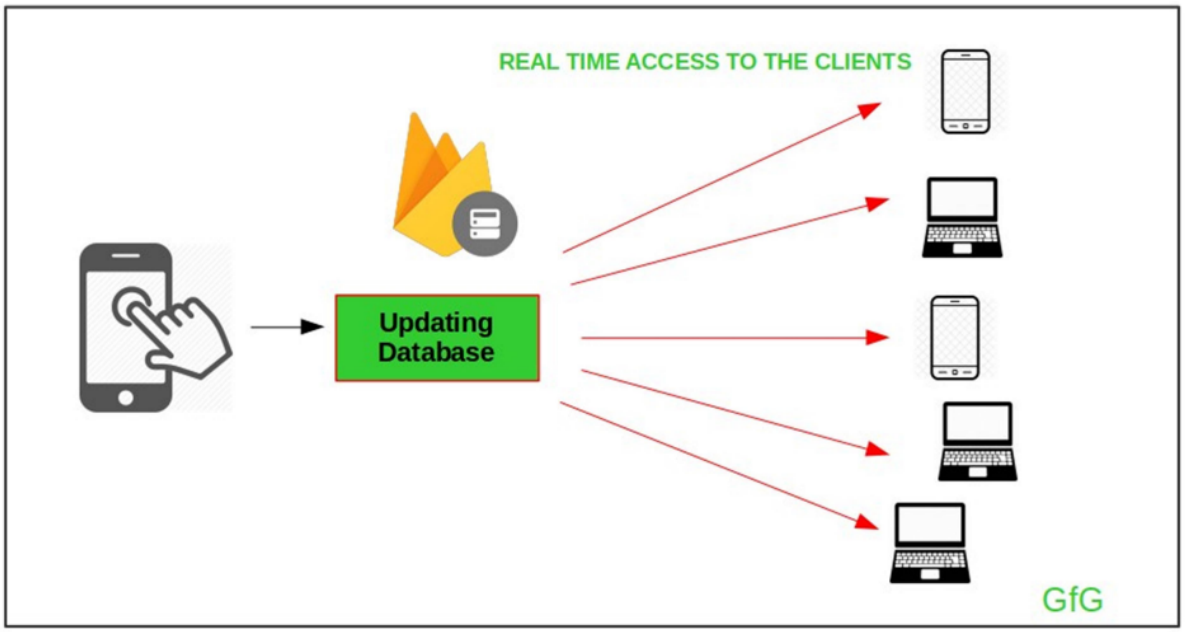
1. Select **Tools > Firebase** to open the **Assistant** window.
2. Click to expand one of the listed features.
3. Click **Get Started with Firebase Analytics** to open a tutorial that connects you to Firebase and adds the necessary code to your app.

Firebase Realtime Database is a Cloud hosted database, i.e. it runs on a cloud and access to the user is provided as a service. It stores data in JSON (Javascript Object Notation) format, a format to store or transport data. All the users connected to it can get access to the data at Real Time.

Features of Firebase Realtime Database?

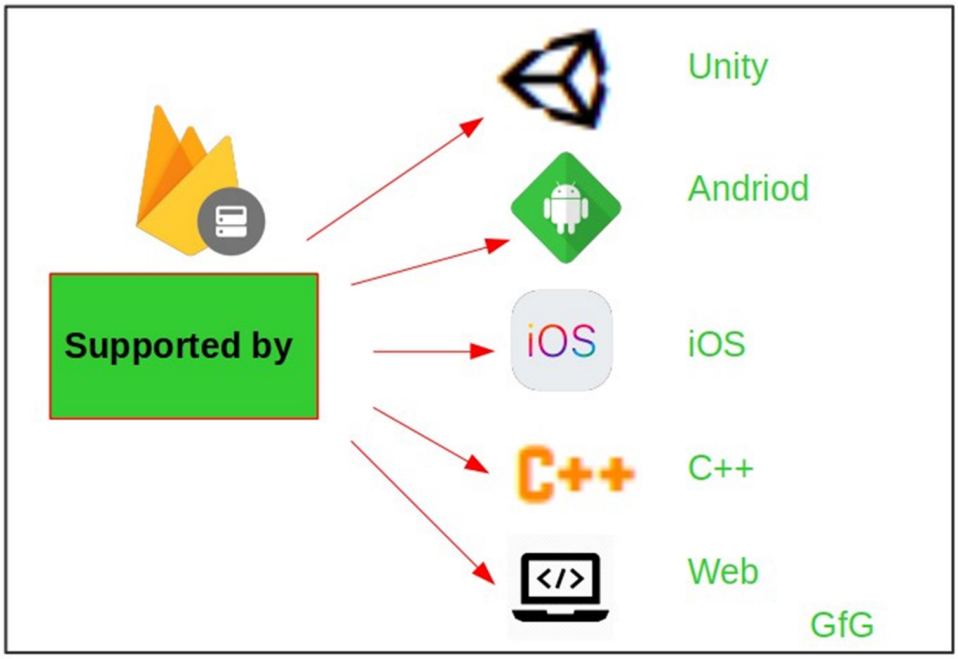


1.Real Time: Due to the Data synchronization used in Real Time, every update is received by the devices/clients in no time.



2.No need of Application Server: As the database can be accessed directly from the mobile device or browser there is no need for an Application Server.

3.Support by various languages and platforms:



4.Splitting Data: The client can split the data across multiple database instances for the same project.

5.Client-Side Code: The dynamic applications with the secured data can be accessed directly from the client-side code.

6.Cross-Platform: It can be used for building a back-end for various platforms like Android, iOS, Web, iOS as well as JavaScript SDK.

Structuring the Realtime Database:

Firebase Realtime Database stores data as one large JSON tree. It stores simple data easily, but the unstructured hierarchical data is difficult to organize. Unlike SQL there are no tables here.

What is JSON Tree Model?

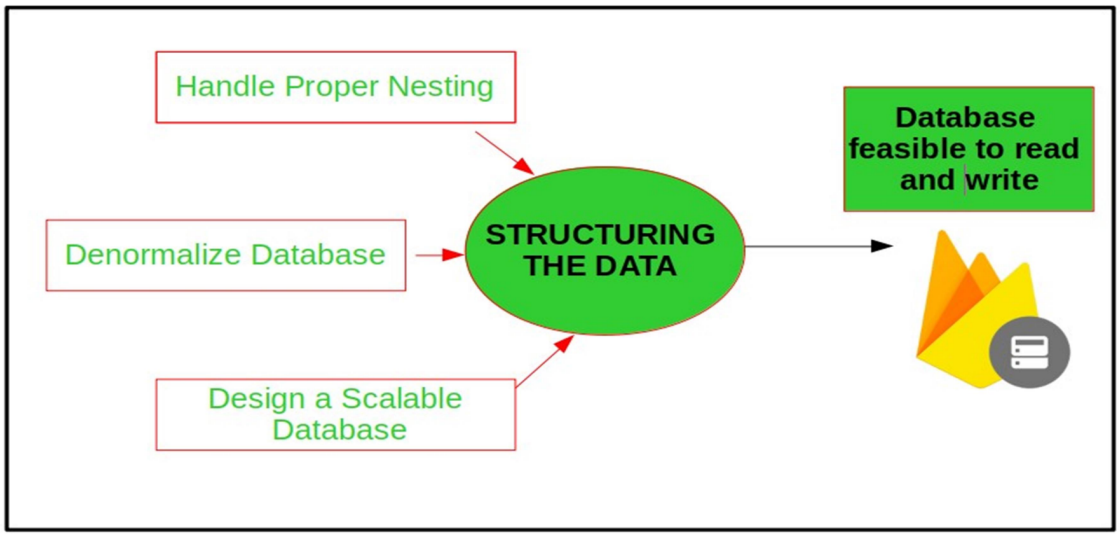
- JSON Tree Model is inspired from JSON(JavaScript Object Notation) used to represent the JSON document which generally consists of key-value pair in memory .

Why to structure the data (What are the advantages of doing it)?

- If data is stored in well formatted structure then it is easy to save as well as retrieve it easily.
- Querying becomes easy on the structured data.
- It becomes feasible to refer the data in structured format.

Key points to remember while structuring the data:

Before writing and reading the data into the database, the main aim of the developer should be constructing the structure of the database.



SOURCE CODE:

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <ListView
```

```
        android:id="@+id/lv_cities"
        android:layout_width="409dp"
        android:layout_height="354dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.289" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
package com.example.exp7_firebase_5c0;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.Map;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.GenericTypeIndicator;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.Map;
public class MainActivity extends AppCompatActivity {
   ListView lv_cities;
   ArrayAdapter<String> aadap;
   ArrayList<String> arr_list;
   protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lv_cities = findViewById(R.id.lv_cities);
        arr_list = new ArrayList<String>(); // Initialize ArrayList before creating ArrayAdapter
        aadap = new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, arr_list);
        DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReference("cities");
        databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.exists()) {
                    Toast.makeText(getApplicationContext(), "hi", Toast.LENGTH_LONG).show();
                    GenericTypeIndicator<Map<String, String>> genericTypeIndicator = new
GenericTypeIndicator<Map<String, String>>() {};
                    Map<String, String> map = snapshot.getValue(genericTypeIndicator);
                    if (map != null) {
                        for (Map.Entry<String, String> map1 : map.entrySet()) {
                            arr_list.add(map1.getValue());
                        }
                        aadap.notifyDataSetChanged(); // Move notifyDataSetChanged() here
                    }
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // Handle onCancelled
            }
        });
        lv_cities.setAdapter(aadap); // Set adapter after adding data to the ArrayList
    }}
```

AndroidManifest.java:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
```

```
        android:supportsRtl="true"
        android:theme="@style/Theme.Exp7_firebase_5c0"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT: