final Button button = (Button) findViewById(R.id.button);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(getApplicationContext(),Main2Activity.class);

startActivity(i);

}

});

**@Entity**

**public** **class** User {

  @PrimaryKey(autoGenerate = **true**)

**private** **int** id;

  @ColumnInfo(name = "first\_name")

**private** String firstName;

**public** **int** getId() {

**return** id;

  }

**public** **void** setId(**int** id) {

**this**.id = id;

  }

**public** String getFirstName() {

**return** firstName;

  }

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

  }

}

**@Dao**

**public** **interface** UserDao {

**@Insert**

  Long insert(User u);

**@Query**("SELECT \* FROM `User` ORDER BY `id` DESC")

  List<User> getAllUsers();

**@Query**("SELECT \* FROM `User` WHERE `id` =:id")

  User getUser(**int** id);

**@Update**

**void** update(User u);

**@Delete**

**void** delete(User u);

}

**@Database(entities = {User.class}, version = 1)**

**public** **abstract** **class** **UserDatabase** **extends** **RoomDatabase**

{

**public** **abstract** UserDao userDao();

**private** **static** UserDatabase INSTANCE;

**public** **static** UserDatabase getAppDatabase(Context context) {

**if** (INSTANCE == **null**) {

      INSTANCE = Room.databaseBuilder(context.getApplicationContext(), UserDatabase.**class**, "user-database").build();

    }

**return** INSTANCE;

  }

**public** **static** **void** destroyInstance() {

    INSTANCE = **null**;

  }

}

1. Insert

|  |
| --- |
| //get the database instance  UserDatabase ud = UserDatabase.getAppDatabase(c.get());    //init the entity  User u = **new** User();  u.setFirstName("John");  u.setLastName("Doe");  u.setPhone("1234567890");  u.setEmail("johndoe@website.com");  u.setAddress("Unknown");    //init dao and perform operation  UserDao dao = ud.userDao();  dao.insert(u); |

1. Update

|  |
| --- |
| //get the database instance  UserDatabase ud = UserDatabase.getAppDatabase(c.get());    //init the entity  User u = **new** User();  u.setId(3);  u.setFirstName("Jane");  u.setLastName("Doe");  u.setPhone("0987654321");  u.setEmail("janedoe@website.com");  u.setAddress("Unknown");    //init dao and perform operation  UserDao dao = ud.userDao();  dao.update(u); |

1. Delete

|  |
| --- |
| //get the database instance  UserDatabase ud = UserDatabase.getAppDatabase(c.get());    //init the entity  User u = **new** User();  u.setId(3);  u.setFirstName("Jane");  u.setLastName("Doe");  u.setPhone("0987654321");  u.setEmail("janedoe@website.com");  u.setAddress("Unknown");    //init dao and perform operation  UserDao dao = ud.userDao();  dao.delete(u); |

1. Retrieve/Read

|  |
| --- |
| //get the database instance  UserDatabase ud = UserDatabase.getAppDatabase(c.get());    //init dao and perform operation  UserDao dao = ud.userDao();    //get all users  List<User> users = dao.getAllUsers();    //get single user by id  User u = dao.getUser(3); |

**Steps to detect basic gestures**

**Step-1 : Create an object of GestureDetector class.**

GestureDetector gestureDetector;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

gestureDetector=new GestureDetector(this, new GestureListener());

}

**Step-2 : Extend another class with GestureDetector.SimpleOnGestureListener to act as a listener and override some methods.**

private class GestureListener extends GestureDetector.SimpleOnGestureListener{

@Override

public boolean onSingleTapUp(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onSingleTapUp() called",Toast.LENGTH\_SHORT).show();

return super.onSingleTapUp(e);

}

@Override

public void onLongPress(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onLongPress() called",Toast.LENGTH\_SHORT).show();

super.onLongPress(e);

}

@Override

public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {

Toast.makeText(getApplicationContext(),"onScroll() called",Toast.LENGTH\_SHORT).show();

return super.onScroll(e1, e2, distanceX, distanceY);

}

@Override

public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {

Toast.makeText(getApplicationContext(),"onFling() called",Toast.LENGTH\_SHORT).show();

return super.onFling(e1, e2, velocityX, velocityY);

}

@Override

public void onShowPress(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onShowPress() called",Toast.LENGTH\_SHORT).show();

super.onShowPress(e);

}

@Override

public boolean onDown(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onDown() called",Toast.LENGTH\_SHORT).show();

return super.onDown(e);

}

@Override

public boolean onDoubleTap(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onDoubleTap() called",Toast.LENGTH\_SHORT).show();

return super.onDoubleTap(e);

}

@Override

public boolean onDoubleTapEvent(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onDoubleTapEvent() called",Toast.LENGTH\_SHORT).show();

return super.onDoubleTapEvent(e);

}

@Override

public boolean onSingleTapConfirmed(MotionEvent e) {

Toast.makeText(getApplicationContext(),"onSingleTapConfirmed() called",Toast.LENGTH\_SHORT).show();

return super.onSingleTapConfirmed(e);

}

}

**Step-3 : Override and implement onTouchEvent() method**

@Override

public boolean onTouchEvent(MotionEvent event) {

gestureDetector.onTouchEvent(event);

return super.onTouchEvent(event);

}

<https://javapapers.com/android/android-chart-example-app-using-mpandroidchart/>

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import com.github.mikephil.charting.charts.BarChart;

import com.github.mikephil.charting.data.BarData;

import com.github.mikephil.charting.data.BarDataSet;

import com.github.mikephil.charting.data.BarEntry;

import com.github.mikephil.charting.interfaces.datasets.IBarDataSet;

import com.github.mikephil.charting.utils.ColorTemplate;

import java.util.ArrayList;

public class BarChartActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bar\_chart);

BarChart chart = findViewById(R.id.barchart);

ArrayList NoOfEmp = new ArrayList();

NoOfEmp.add(new BarEntry(945f, 0));

NoOfEmp.add(new BarEntry(1040f, 1));

NoOfEmp.add(new BarEntry(1133f, 2));

ArrayList year = new ArrayList();

year.add("2008");

year.add("2009");

year.add("2010");

BarDataSet bardataset = new BarDataSet(NoOfEmp, "No Of Employee");

chart.animateY(5000);

BarData data = new BarData(year, bardataset);

bardataset.setColors(ColorTemplate.COLORFUL\_COLORS);

chart.setData(data);

}

}

package com.example.sensortutorial;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.os.Bundle;

import android.view.View;

import android.widget.TextView;

import java.util.List;

public class MainActivity extends AppCompatActivity implements SensorEventListener {

    private SensorManager mgr;

    private TextView txtList;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        mgr = (SensorManager)getSystemService(Context.SENSOR\_SERVICE);

        txtList = (TextView)findViewById(R.id.sensorslist);

        List<Sensor> sensorList = mgr.getSensorList(Sensor.TYPE\_ALL);

        StringBuilder strBuilder = new StringBuilder();

        for(Sensor s: sensorList){

            strBuilder.append(s.getType()+"----"+s.getName()+"\n");

        }

        txtList.setVisibility(View.VISIBLE);

        txtList.setText(strBuilder);

    }

    @Override

    public void onSensorChanged(SensorEvent event) {

    }

    @Override

    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }

}

package com.example.sensortutorial;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.os.Bundle;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

public class SensorTypeActivity extends AppCompatActivity implements SensorEventListener {

    private TextView textView;

    private SensorManager sensorManager;

    private Sensor accelerometerSensor;

    private Sensor proximitySensor;

    private Sensor lightSensor;

    private Sensor stepCounterSensor;

    private Sensor tempSensor;

    private Sensor gyroscopeSensor;

    private int currentSensor;

    private long lastUpdate = 0;

    private float last\_x, last\_y, last\_z;

    private static final int SHAKE\_THRESHOLD = 600;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_sensor\_type);

        textView = findViewById(R.id.tvResult);

        sensorManager = (SensorManager) getSystemService(Context.SENSOR\_SERVICE);

        accelerometerSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER);

        proximitySensor = sensorManager.getDefaultSensor(Sensor.TYPE\_PROXIMITY);

        lightSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_LIGHT);

        stepCounterSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_STEP\_DETECTOR);

        gyroscopeSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_GYROSCOPE);

        tempSensor = sensorManager.getDefaultSensor(Sensor.TYPE\_AMBIENT\_TEMPERATURE);

    }

    public boolean checkSensorAvailability(int sensorType) {

        boolean isSensor = false;

        if (sensorManager.getDefaultSensor(sensorType) != null) {

            isSensor = true;

        }

        return isSensor;

    }

    @Override

    public void onSensorChanged(SensorEvent event) {

        if (event.sensor.getType() == currentSensor) {

        //    Toast.makeText(getApplicationContext(),event.sensor.getType(),Toast.LENGTH\_LONG).show();

            if (currentSensor == Sensor.TYPE\_LIGHT) {

                float valueZ = event.values[0];

                textView.setText("Brightness " + valueZ);

            } else if (currentSensor == Sensor.TYPE\_PROXIMITY) {

                float distance = event.values[0];

                textView.setText("Proximity " + distance);

            } else if (currentSensor == Sensor.TYPE\_STEP\_DETECTOR) {

                float steps = event.values[0];

                textView.setText("Steps : " + steps);

            } else if (currentSensor == Sensor.TYPE\_ACCELEROMETER) {

                float x = event.values[0];

                float y = event.values[1];

                float z = event.values[2];

                long curTime = System.currentTimeMillis();

                if ((curTime - lastUpdate) > 100) {

                    long diffTime = (curTime - lastUpdate);

                    lastUpdate = curTime;

                    float speed = Math.abs(x + y + z - last\_x - last\_y - last\_z) / diffTime \* 10000;

                    if (speed > SHAKE\_THRESHOLD) {

                        Toast.makeText(getApplicationContext(), "Your phone just shook", Toast.LENGTH\_LONG).show();

                    }

                    last\_x = x;

                    last\_y = y;

                    last\_z = z;

                }

            } else if (currentSensor == Sensor.TYPE\_GYROSCOPE) {

                if (event.values[2] > 0.5f) {

                    textView.setText("Anti Clock");

                } else if (event.values[2] < -0.5f) {

                    textView.setText("Clock");

                }

            } else if (currentSensor == Sensor.TYPE\_AMBIENT\_TEMPERATURE) {

                textView.setText("Ambient Temp in Celsius :" + event.values[0]);

            }

        }

    }

    @Override

    public void onAccuracyChanged(Sensor sensor, int accuracy) {

    }

    public void accelerometerSensorOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_ACCELEROMETER)) {

            currentSensor = Sensor.TYPE\_ACCELEROMETER;

        }else {

            textView.setText("Accelerometer not available");

        }

    }

    public void proximitySensorOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_PROXIMITY)) {

            currentSensor = Sensor.TYPE\_PROXIMITY;

        }else {

            textView.setText("Proximity Sensor not available");

        }

    }

    public void gyroscopeSensorOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_GYROSCOPE)) {

            currentSensor = Sensor.TYPE\_GYROSCOPE;

        } else {

            textView.setText("Gyroscope Sensor not available");

        }

    }

    public void lightSensorOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_LIGHT)) {

            currentSensor = Sensor.TYPE\_LIGHT;

        } else {

            textView.setText("Light Sensor not available");

        }

    }

    public void stepCounterOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_STEP\_DETECTOR)) {

            currentSensor = Sensor.TYPE\_STEP\_DETECTOR;

        } else {

            textView.setText("Step Counter Sensor not available");

        }

    }

    public void ambientTempSensorOnClick(View view) {

        if (checkSensorAvailability(Sensor.TYPE\_AMBIENT\_TEMPERATURE)) {

            currentSensor = Sensor.TYPE\_AMBIENT\_TEMPERATURE;

        } else {

            textView.setText("Ambient Temperature Sensor not available");

        }

    }

    @Override

    protected void onResume() {

        super.onResume();

        sensorManager.registerListener(this, accelerometerSensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

        sensorManager.registerListener(this, lightSensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

        sensorManager.registerListener(this, proximitySensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

        sensorManager.registerListener(this, stepCounterSensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

        sensorManager.registerListener(this, tempSensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

        sensorManager.registerListener(this, gyroscopeSensor,

                SensorManager.SENSOR\_DELAY\_NORMAL);

    }

    @Override

    protected void onPause() {

        super.onPause();

        sensorManager.unregisterListener(this);

    }

}

package com.example.sensortutorial;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;

import android.hardware.Sensor;

import android.hardware.SensorEvent;

import android.hardware.SensorEventListener;

import android.hardware.SensorManager;

import android.os.Bundle;

import android.view.View;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

public class ChangeBGActivity extends AppCompatActivity implements SensorEventListener {

    private SensorManager sensorManager;

    private boolean isColor = false;

    private TextView view;

    private RelativeLayout layout;

    private long lastUpdate;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_change\_b\_g);

        view = (TextView) findViewById(R.id.textView);

        layout = (RelativeLayout) findViewById(R.id.rellayout);

        layout.setBackgroundColor(Color.BLUE);

        sensorManager = (SensorManager) getSystemService(SENSOR\_SERVICE);

        lastUpdate = System.currentTimeMillis();

    }

    //overriding two methods of SensorEventListener

    @Override

    public void onAccuracyChanged(Sensor sensor, int accuracy) {}

    @Override

    public void onSensorChanged(SensorEvent event) {

        if (event.sensor.getType() == Sensor.TYPE\_ACCELEROMETER) {

            getAccelerometer(event);

        }

    }

    private void getAccelerometer(SensorEvent event) {

        float[] values = event.values;

        // Movement

        float x = values[0];

        float y = values[1];

        float z = values[2];

        float accelationSquareRoot = (x \* x + y \* y + z \* z)

                / (SensorManager.GRAVITY\_EARTH \* SensorManager.GRAVITY\_EARTH);

        long actualTime = System.currentTimeMillis();

        Toast.makeText(getApplicationContext(),String.valueOf(accelationSquareRoot)+" "+

                SensorManager.GRAVITY\_EARTH,Toast.LENGTH\_SHORT).show();

        if (accelationSquareRoot >= 2) //it will be executed if you shuffle

        {

            if (actualTime - lastUpdate < 200) {

                return;

            }

            lastUpdate = actualTime;//updating lastUpdate for next shuffle

            if (isColor) {

                layout.setBackgroundColor(Color.BLUE);

            } else {

                layout.setBackgroundColor(Color.CYAN);

            }

            isColor = !isColor;

        }

    }

    @Override

    protected void onResume() {

        super.onResume();

        // register this class as a listener for the orientation and

        // accelerometer sensors

        sensorManager.registerListener(this,sensorManager.getDefaultSensor(Sensor.TYPE\_ACCELEROMETER),

                SensorManager.SENSOR\_DELAY\_NORMAL);

    }

    @Override

    protected void onPause() {

        // unregister listener

        super.onPause();

        sensorManager.unregisterListener(this);

    }

}

<https://www.geeksforgeeks.org/adding-firebase-to-android-app/>

Create a new project in the firebase by clicking on the Add project.

Now open the android studio and click on Tools in the upper left corner.

Now click on the Firebase option in the drop down menu.

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

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

Add the dependencies of your service by clicking on the Add [YOUR SERVICE NAME] to the app option. (In the image below, the Firebase cloud messaging service is chosen)

Manually adding firebase

In this, the steps involve:

Create a firebase project

Create a project by clicking on create project in the firebase console.

Fill the necessary details in the pop up window about the project. Edit the project ID if required.

Click on create project to finally create it.

Now add this project to the android app

Click on the Add firebase to your android app option on the starting window.

A prompt will open where to enter the package name of the app.

Now the app is connected to the Firebase. Now all the cloud based as well server based services can be easily used in the app.

Now the app will be registered with firebase.

Also, the SHA1 certificate, can be given, of the app by following steps:

Go to android studio project

↳ gradle

↳ root folder

↳ Tasks

↳ Android

↳ signingReport

↳ copy paste SHA1 from console

Now download the google-services.json file and place it in the root directory of the android app.

Now add the following in the project.

Adding the sdk in the project.

Add the following code to the PROJECT-LEVELbuild.gradle of the app.

buildscript {

dependencies {

classpath 'com.google.gms:google-services:4.0.0'

}

}

Add the following code to APP-LEVEL build.gradle of the app.

dependencies {

compile 'com.google.firebase:firebase-core:16.0.0'

}

...

// Add to the bottom of the file

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

Now Sync the gradle by clicking on sync now.

After adding the above code(sdk), run the app to send the verification to the Firebase console.