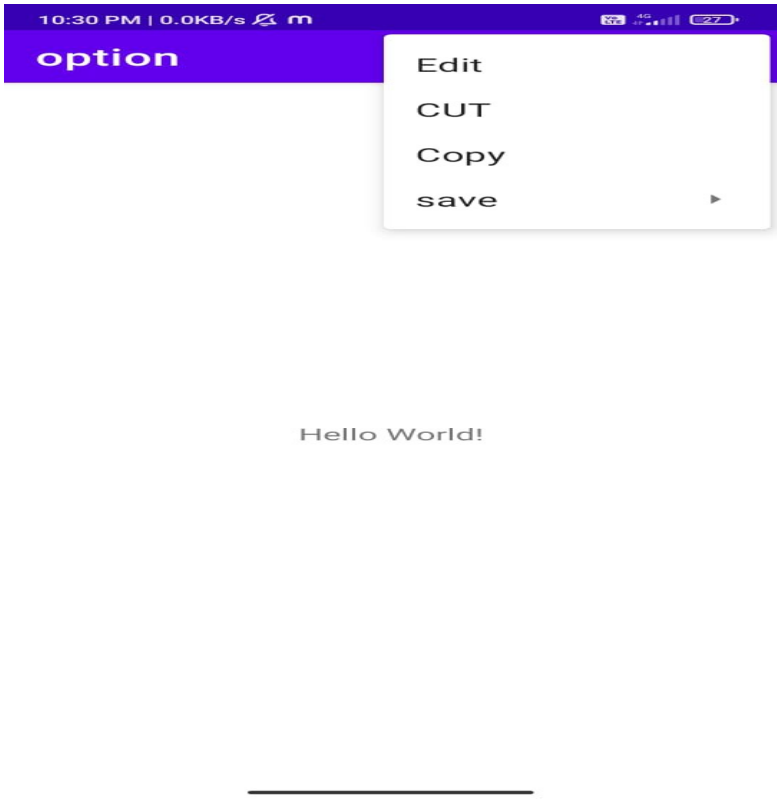
Option menu

optionmenu
```
 package com.example.optionmenu

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast

   class MainActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

   }

      override fun onCreateOptionsMenu(menu: Menu?): Boolean {
         menuInflater.inflate(R.menu.option_menu,menu)
         return super.onCreateOptionsMenu(menu)

      }

      override fun onOptionsItemSelected(item: MenuItem): Boolean {
         when(item.itemId){
            R.id.paste -> Toast.makeText(this ,"paste is selected",Toast.LENGTH_LONG).show()
            R.id.copy -> Toast.makeText(this, "copy the file",Toast.LENGTH_LONG).show()

      }
         return super.onOptionsItemSelected(item)
      }

}
```

option_menu.xml(ADD NEW REOURCE FILE)
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

   <item
      android:id="@+id/copy"
      android:title="copy" />
   <item
      android:id="@+id/paste"
      android:title="paste" />
   </menu >
```

output



practical 12-database application

database app
mainactivity.kt

```
package com.example.databaseapp

import android.annotation.SuppressLint
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import java.util.jar.Attributes.Name

class MainActivity : AppCompatActivity() {




    @SuppressLint("Range")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
val addName=findViewById<Button>(R.id.addName)
        val enterName=findViewById<TextView>(R.id.enterName)
        val enterAge=findViewById<TextView>(R.id.enterAge)
        val Name=findViewById<TextView>(R.id.Name)
        val Age=findViewById<TextView>(R.id.Age)

        // below code is to add on click
        // listener to our add name button
       addName.setOnClickListener {

            // below we have created
            // a new DBHelper class,
            // and passed context to it
            val db = DBHelper(this,null)

            // creating variables for values
            // in name and age edit texts
            val name = enterName.text.toString()
            val age = enterAge.text.toString()

            // calling method to add
            // name to our database
            db.addName(name, age)

            // Toast to message on the screen
```

```
            Toast.makeText(this, name + " added to database", Toast.LENGTH_LONG).show()

            // at last, clearing edit texts

        }

        // below code is to add on click
        // listener to our print name button
        val printName=findViewById<Button>(R.id.printName)
        printName.setOnClickListener {

            // creating a DBHelper class
            // and passing context to it
            val db = DBHelper(this, null)

            // below is the variable for cursor
            // we have called method to get
            // all names from our database
            // and add to name text view
            val cursor = db.getName()

            // moving the cursor to first position and
            // appending value in the text view
            cursor!!.moveToFirst()
            Name.append(cursor.getString(cursor.getColumnIndex(DBHelper.NAME_COl)) + "\n")
            Age.append(cursor.getString(cursor.getColumnIndex(DBHelper.AGE_COL)) + "\n")

            // moving our cursor to next
            // position and appending values
            while (cursor.moveToNext()) {
                Name.append(cursor.getString(cursor.getColumnIndex(DBHelper.NAME_COl)) + "\n")
                Age.append(cursor.getString(cursor.getColumnIndex(DBHelper.AGE_COL)) + "\n")
            }

            // at last we close our cursor
            cursor.close()// below code is to add on click
            // listener to our add name button

        }
    }
}
```

DBHelper.kt(add it using kotlin class)

```
package com.example.databaseapp

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
```

```
import android.database.sqlite.SQLiteOpenHelper

class DBHelper(context: Context, factory: SQLiteDatabase.CursorFactory?) :
    SQLiteOpenHelper(context, DATABASE_NAME, factory, DATABASE_VERSION) {

    // below is the method for creating a database by a sqlite query
    override fun onCreate(db: SQLiteDatabase) {
        // below is a sqlite query, where column names
        // along with their data types is given
        val query = ("CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY, " +
                NAME_COl + " TEXT," +
                AGE_COL + " TEXT" + ")")

        // we are calling sqlite
        // method for executing our query
        db.execSQL(query)
    }

    override fun onUpgrade(db: SQLiteDatabase, p1: Int, p2: Int) {
        // this method is to check if table already exists
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME)
        onCreate(db)
    }

    // This method is for adding data in our database
    fun addName(name : String, age : String ){

        // below we are creating
        // a content values variable
        val values = ContentValues()

        // we are inserting our values
        // in the form of key-value pair
        values.put(NAME_COl, name)
        values.put(AGE_COL, age)

        // here we are creating a
        // writable variable of
        // our database as we want to
        // insert value in our database
        val db = this.writableDatabase

        // all values are inserted into database
        db.insert(TABLE_NAME, null, values)

        // at last we are
        // closing our database
        db.close()
```

```
    }

    // below method is to get
    // all data from our database
    fun getName(): Cursor? {

        // here we are creating a readable
        // variable of our database
        // as we want to read value from it
        val db = this.readableDatabase

        // below code returns a cursor to
        // read data from the database
        return db.rawQuery("SELECT * FROM " + TABLE_NAME, null)

    }

    companion object{
        // here we have defined variables for our database

        // below is variable for database name
        private val DATABASE_NAME = "GEEKS_FOR_GEEKS"

        // below is the variable for database version
        private val DATABASE_VERSION = 1

        // below is the variable for table name
        val TABLE_NAME = "gfg_table"

        // below is the variable for id column
        val ID_COL = "id"

        // below is the variable for name column
        val NAME_COl = "name"

        // below is the variable for age column
        val AGE_COL = "age"
    }
}
```

manifest.xml

```
<?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.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>


    <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:supportsRtl="true"
        android:theme="@style/Theme.DATABASEAPP"
        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>
```

implicit_intent app(i.e there are two pages/activity in app)
mainactivity.kt

```
package com.example.implicit_intent

import android.content.Intent
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        var bottan =findViewById<Button>(R.id.button)
        bottan.setOnClickListener(){
        val i=Intent(applicationContext,MainActivity2::class.java)
        startActivity(i)
    }}
}
```

mainactivity2.kt

```
package com.example.implicit_intent

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
```

```
import android.os.Bundle
import android.widget.Button

class MainActivity2 : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main2)

        var bottan = findViewById<Button>(R.id.button2)
        bottan.setOnClickListener() {
            val i = Intent(applicationContext, MainActivity::class.java)
            startActivity(i)
        }
    }
}
```

there are two design files in this application

output: