

Running Your Application

Now you can run your Android code either on your Android device or through Android Emulator. I will be telling you to run your application on Android Emulator for this you need to set up your Android Virtual Device that would run your code , now to set up a Android Virtual Device or AVD you need to create one using the Android Virtual Device Manager or the AVD Manager, in order to do so follow these steps –

1. Open up the AVD Manager.
2. In the Android Virtual Device Manager panel, click New.
3. Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
4. Click Create AVD.
5. Select the new AVD from the Android Virtual Device Manager and click Start.
6. After the emulator boots up, unlock the emulator screen.

Android Programs

Program-1 Write an android program to print Hello World

MainActivity.kt

```
package com.example.helloworld

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.activity.enableEdgeToEdge
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.helloworld.ui.theme.HelloWorldTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            HelloWorldTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    Greeting(
                        name = "World",
                        modifier = Modifier.padding(innerPadding)
                    )
                }
            }
        }
    }
}
```

```

}

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier
    )
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    HelloWorldTheme {
        Greeting("World")
    }
}

```

layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

</LinearLayout>

```

Program-2 Write an android program for button.

MainActivity.kt

```

package com.example.firstapp

import android.os.Bundle
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.view.View

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {

        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,

```

```

systemBars.bottom)
    insets
    }
}

fun onSubmit(view: View){

    Toast.makeText(this,"You have clicked clickme
Button",Toast.LENGTH_LONG).show()

}

}

```

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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <Button
        android:id="@+id/btn1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onSubmit"
        android:text="CLICKME"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.459" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Program-3 Write a program for Edittext

MainActivity.kt

```

package com.example.demoapp

import android.os.Bundle
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import android.view.View
import android.widget.EditText
import android.widget.Toast

class MainActivity : AppCompatActivity() {

```

```

lateinit var editText: EditText
lateinit var string: String

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    enableEdgeToEdge()
    setContentView(R.layout.activity_main)
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) {
v, insets ->
        val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
        v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom)
        insets
    }
}

fun onSubmit(view: View) {

    editText= findViewById(R.id.EditTextName)
    string = editText.text.toString()

    Toast.makeText(this,string,Toast.LENGTH_LONG).show()

}
}

```

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:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <EditText
        android:id="@+id/EditTextName"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:ems="10"
        android:inputType="text"
        android:hint="Enter your text here"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.442"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.144" />

```

```

<Button
    android:id="@+id/BtnSubmit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onSubmit"
    android:text="Submit"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.284" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Program-4 Write a program for arithmetic app.

MainActivity.kt

```

package com.example.arithmeticapp

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import com.example.calculationapp.R

class MainActivity : AppCompatActivity() {

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

        // Get references to UI elements
        val num1EditText = findViewById<EditText>(R.id.num1)
        val num2EditText = findViewById<EditText>(R.id.num2)
        val resultTextView = findViewById<TextView>(R.id.result)

        // Define the buttons for operations
        val addButton = findViewById<Button>(R.id.addButton)
        val subButton = findViewById<Button>(R.id.subButton)
        val mulButton = findViewById<Button>(R.id.mulButton)
        val divButton = findViewById<Button>(R.id.divButton)

        // Set click listeners for buttons
        addButton.setOnClickListener {
            performOperation(num1EditText, num2EditText, resultTextView,
"add")
        }

        subButton.setOnClickListener {
            performOperation(num1EditText, num2EditText, resultTextView,
"subtract")
        }

        mulButton.setOnClickListener {
            performOperation(num1EditText, num2EditText, resultTextView,

```

```

"multiply")
    }

    divButton.setOnClickListener {
        performOperation(num1EditText, num2EditText, resultTextView,
"divide")
    }
}

// Function to perform operations
private fun performOperation(num1EditText: EditText, num2EditText:
EditText, resultTextView: TextView, operation: String) {
    // Get the text from EditTexts
    val num1 = num1EditText.text.toString().toDoubleOrNull()
    val num2 = num2EditText.text.toString().toDoubleOrNull()

    if (num1 != null && num2 != null) {
        var result = 0.0
        when (operation) {
            "add" -> result = num1 + num2
            "subtract" -> result = num1 - num2
            "multiply" -> result = num1 * num2
            "divide" -> if (num2 != 0.0) result = num1 / num2 else
resultTextView.text = "Cannot divide by zero"
        }
        resultTextView.text = "Result: $result"
    } else {
        resultTextView.text = "Please enter valid numbers"
    }
}
}
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Adding padding -->

    <EditText
        android:id="@+id/num1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:hint="Enter first number"
        android:inputType="numberDecimal"
        android:padding="16dp"
        android:textSize="18sp" />

    <EditText

```

```

        android:id="@+id/num2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:hint="Enter second number"
        android:inputType="numberDecimal"
        android:padding="16dp"
        android:textSize="18sp" /> <!-- Adding padding -->

<TextView
    android:id="@+id/result"
    android:layout_width="match_parent"
    android:layout_height="63dp"
    android:layout_marginTop="16dp"
    android:text="Result: "
    android:textAlignment="center"
    android:textSize="20sp" />

<Button
    android:id="@+id/addButton"
    android:layout_width="141dp"
    android:layout_height="wrap_content"
    android:text="Add"
    android:textSize="18sp" />

<Button
    android:id="@+id/subButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:text="Subtract"
    android:textSize="18sp" />

<Button
    android:id="@+id/mulButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:text="Multiply"
    android:textSize="18sp" />

<Button
    android:id="@+id/divButton"
    android:layout_width="104dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:text="Divide"
    android:textSize="18sp" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:gravity="center"
    android:orientation="horizontal">

```

```
</RelativeLayout>

</LinearLayout>
```

Program-5 : Write a program for ImageView.

MainActivity.kt

```
package com.example.imageviewapp

import android.os.Bundle
import android.widget.ImageView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

        // Use findViewById to get the ImageView
        val imageView: ImageView = findViewById(R.id.imageView)

        // Set the star image from the system drawable
        imageView.setImageResource(android.R.drawable.btn_star_big_on)
    }
}
```

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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="268dp"
        android:layout_height="326dp"
        android:layout_marginStart="100dp"
        android:layout_marginTop="48dp"
        android:contentDescription="Star"
        android:src="@android:drawable/btn_star_big_on"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```


Program-6 Write a program for option button.

MainActivity.kt

```
package com.example.radiobuttonapp

import android.os.Bundle
import android.widget.RadioButton
import android.widget.RadioGroup
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
//import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {

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

        // Find the RadioGroup and TextView by their IDs
        val radioGroup: RadioGroup = findViewById(R.id.radioGroup)
        val resultText: TextView = findViewById(R.id.resultText)

        // Set a listener for the RadioGroup
        radioGroup.setOnCheckedChangeListener { group, checkedId ->
            // Get the selected RadioButton by ID
            val selectedRadioButton: RadioButton = findViewById(checkedId)
            // Set the text of the TextView to show the selected option
            resultText.text = "Selected: ${selectedRadioButton.text}"
        }
    }
}
```

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">

    <!-- RadioGroup to group RadioButtons -->
    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:layout_centerInParent="true">
```

```

<!-- First RadioButton -->
<RadioButton
    android:id="@+id/radioButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Option 1" />

<!-- Second RadioButton -->
<RadioButton
    android:id="@+id/radioButton2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Option 2" />

</RadioGroup>

<!-- TextView to display the selected radio button -->
<TextView
    android:id="@+id/resultText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select an option"
    android:layout_marginTop="32dp"
    app:layout_constraintTop_toBottomOf="@+id/radioGroup"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Program-7 Write a program for Listview.

MainActivity.kt

```

package com.example.listviewapp

import android.os.Bundle
import android.widget.ArrayAdapter
import android.widget.ListView
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

        // List of data to display in ListView
        val items = arrayOf("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")

        // Find the ListView by ID
        val listView: ListView = findViewById(R.id.listView)

        // Create an ArrayAdapter to bind the data to the ListView
        val adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1,
items)

```

```

        // Set the adapter to the ListView
        listView.adapter = adapter
    }
}

```

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 to display the items -->
    <ListView
        android:id="@+id/listView"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="16dp"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

Program-7 Write a program for progress bar.

MainActivity.kt

```
package com.example.progressbarexample
```

```

import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.widget.Button
import android.widget.ProgressBar
import androidx.appcompat.app.AppCompatActivity

```

```
class MainActivity : AppCompatActivity() {
```

```

    private lateinit var progressBar: ProgressBar
    private lateinit var progressBarIndeterminate: ProgressBar
    private lateinit var startButton: Button
    private var progressStatus = 0
    private val handler = Handler(Looper.getMainLooper())

```

```

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

```

```

    // Initialize UI elements
    progressBar = findViewById(R.id.progressBar)
    progressBarIndeterminate = findViewById(R.id.progressBarIndeterminate)
    startButton = findViewById(R.id.startButton)

```

```

// Start button click listener
startButton.setOnClickListener {
    progressStatus = 0
    progressBar.progress = 0
    progressBarIndeterminate.visibility = ProgressBar.VISIBLE // Show indeterminate progress bar

    Thread {
        while (progressStatus < 100) {
            progressStatus += 5
            handler.post {
                progressBar.progress = progressStatus
            }
            Thread.sleep(500)
        }
        handler.post {
            progressBarIndeterminate.visibility = ProgressBar.INVISIBLE // Hide indeterminate progress bar
        }
    }.start()
}
}
}

```

Activity_main.xml

```
package com.example.progressbarexample
```

```

import android.os.Bundle
import android.os.Handler
import android.os.Looper
import android.widget.Button
import android.widget.ProgressBar
import androidx.appcompat.app.AppCompatActivity

```

```

class MainActivity : AppCompatActivity() {

    private lateinit var progressBar: ProgressBar
    private lateinit var progressBarIndeterminate: ProgressBar
    private lateinit var startButton: Button
    private var progressStatus = 0
    private val handler = Handler(Looper.getMainLooper())

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

        // Initialize UI elements
        progressBar = findViewById(R.id.progressBar)
        progressBarIndeterminate = findViewById(R.id.progressBarIndeterminate)
        startButton = findViewById(R.id.startButton)

        // Start button click listener
        startButton.setOnClickListener {
            progressStatus = 0
            progressBar.progress = 0
            progressBarIndeterminate.visibility = ProgressBar.VISIBLE // Show indeterminate progress bar

```

```

Thread {
    while (progressStatus < 100) {
        progressStatus += 5
        handler.post {
            progressBar.progress = progressStatus
        }
        Thread.sleep(500)
    }
    handler.post {
        progressBarIndeterminate.visibility = ProgressBar.INVISIBLE // Hide indeterminate progress bar
    }
}.start()
}
}
}

```

Program-8 Write a program for toggle button.

MainActivity.kt

```
package com.example.togglebuttonexample
```

```

import android.os.Bundle
import android.widget.TextView
import android.widget.ToggleButton
import androidx.appcompat.app.AppCompatActivity

```

```

class MainActivity : AppCompatActivity() {

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

        // Initialize UI elements
        val toggleButton = findViewById<ToggleButton>(R.id.toggleButton)
        val statusTextView = findViewById<TextView>(R.id.statusTextView)

        // Set Toggle Button change listener
        toggleButton.setOnCheckedChangeListener { _, isChecked ->
            if (isChecked) {
                statusTextView.text = "Toggle is ON"
            } else {
                statusTextView.text = "Toggle is OFF"
            }
        }
    }
}

```

Activity_main.xml

```

package com.example.togglebuttonexample

import android.os.Bundle
import android.widget.TextView
import android.widget.ToggleButton
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

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

        // Initialize UI elements
        val toggleButton = findViewById<ToggleButton>(R.id.toggleButton)
        val statusTextView = findViewById<TextView>(R.id.statusTextView)

        // Set Toggle Button change listener
        toggleButton.setOnCheckedChangeListener { _, isChecked ->
            if (isChecked) {
                statusTextView.text = "Toggle is ON"
            } else {
                statusTextView.text = "Toggle is OFF"
            }
        }
    }
}

```

Program-9 Write a program for login form.

MainActivity.kt

```

package com.example.loginformapp

import android.annotation.SuppressLint
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var usernameEditText: EditText
    private lateinit var passwordEditText: EditText
    private lateinit var loginButton: Button

    @SuppressLint("MissingInflatedId")

```

```

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

    // Initialize views
    usernameEditText = findViewById(R.id.usernameEditText)
    passwordEditText = findViewById(R.id.passwordEditText)
    loginButton = findViewById(R.id.loginButton)

    // Set up login button click listener
    loginButton.setOnClickListener {
        val username = usernameEditText.text.toString()
        val password = passwordEditText.text.toString()

        // Check if both fields are not empty
        if (username.isEmpty() || password.isEmpty()) {
            Toast.makeText(
                this@MainActivity,
                "Please enter both username and password",
                Toast.LENGTH_SHORT
            ).show()
        } else {
            // Display login information
            Toast.makeText(
                this@MainActivity,
                "Username: $username\nPassword: $password",
                Toast.LENGTH_LONG
            ).show()
        }
    }
}

```

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">

    <!-- Username EditText -->
    <EditText
        android:id="@+id/usernameEditText"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="100dp"

```

```

    android:layout_marginEnd="16dp"
    android:layout_marginBottom="24dp"
    android:hint="Username"
    android:inputType="text"
    app:layout_constraintBottom_toTopOf="@+id/passwordEditText"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

<!-- Password EditText -->

<EditText

```

    android:id="@+id/passwordEditText"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="18dp"
    android:hint="Password"
    android:inputType="textPassword"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/usernameEditText" />

```

<!-- Login Button -->

<Button

```

    android:id="@+id/loginButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Login"
    android:textAllCaps="false"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

</androidx.constraintlayout.widget.ConstraintLayout>

Program-10 Write a program for database CRUD operations using sqlite.

Databasehelper.kt

```

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) : SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

    companion object {
        const val DATABASE_NAME = "mydatabase.db"
        const val DATABASE_VERSION = 1
        const val TABLE_NAME = "users"
        const val COLUMN_ID = "id"
        const val COLUMN_NAME = "name"
        const val COLUMN_CITY = "city"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val createTableSQL = """
            CREATE TABLE $TABLE_NAME (
                $COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT,
                $COLUMN_NAME TEXT,
                $COLUMN_CITY TEXT
            )
        """.trimIndent()
        db?.execSQL(createTableSQL)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }

    // Insert record
    fun insertRecord(name: String, city: String): Long {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_NAME, name)
            put(COLUMN_CITY, city)
        }
        return db.insert(TABLE_NAME, null, values)
    }

    // Update record
    fun updateRecord(id: Int, name: String, city: String): Int {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COLUMN_NAME, name)
            put(COLUMN_CITY, city)
        }
        return db.update(TABLE_NAME, values, "$COLUMN_ID = ?", arrayOf(id.toString()))
    }

    // Select all records
    fun getAllRecords(): Cursor {

```

```

        val db = readableDatabase
        return db.rawQuery("SELECT * FROM $TABLE_NAME", null)
    }

    // Delete record by ID
    fun deleteRecord(id: Int): Int {
        val db = writableDatabase
        return db.delete(TABLE_NAME, "$COLUMN_ID = ?", arrayOf(id.toString()))
    }
}

```

MainActivity.kt

```
package com.example.demosqlitesem6
```

```

import DBHelper
import android.annotation.SuppressLint
import android.database.Cursor
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat

```

```

class MainActivity : AppCompatActivity() {

    private lateinit var dbHelper: DBHelper
    private lateinit var nameEditText: EditText
    private lateinit var cityEditText: EditText
    private lateinit var idEditText: EditText
    private lateinit var insertButton: Button
    private lateinit var updateButton: Button
    private lateinit var deleteButton: Button
    private lateinit var selectButton: Button

    @SuppressLint("Range")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        dbHelper = DBHelper(this)

        nameEditText = findViewById(R.id.nameEditText)
        cityEditText = findViewById(R.id.cityEditText)
        idEditText = findViewById(R.id.idEditText)
        insertButton = findViewById(R.id.insertButton)
        updateButton = findViewById(R.id.updateButton)

```

```
deleteButton = findViewById(R.id.deleteButton)
selectButton = findViewById(R.id.selectButton)
```

```
// Insert record
```

```
insertButton.setOnClickListener {
    val name = nameEditText.text.toString()
    val city = cityEditText.text.toString()

    if (name.isNotEmpty() && city.isNotEmpty()) {
        val result = dbHelper.insertRecord(name, city)
        if (result != -1L) {
            Toast.makeText(this, "Record inserted", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(this, "Failed to insert record", Toast.LENGTH_SHORT).show()
        }
    } else {
        Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show()
    }
}
```

```
// Update record
```

```
updateButton.setOnClickListener {
    val id = idEditText.text.toString().toIntOrNull()
    val name = nameEditText.text.toString()
    val city = cityEditText.text.toString()

    if (id != null && name.isNotEmpty() && city.isNotEmpty()) {
        val result = dbHelper.updateRecord(id, name, city)
        if (result > 0) {
            Toast.makeText(this, "Record updated", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(this, "Failed to update record", Toast.LENGTH_SHORT).show()
        }
    } else {
        Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show()
    }
}
```

```
// Delete record
```

```
deleteButton.setOnClickListener {
    val id = idEditText.text.toString().toIntOrNull()
    if (id != null) {
        val result = dbHelper.deleteRecord(id)
        if (result > 0) {
            Toast.makeText(this, "Record deleted", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(this, "Failed to delete record", Toast.LENGTH_SHORT).show()
        }
    } else {
        Toast.makeText(this, "Please enter a valid ID", Toast.LENGTH_SHORT).show()
    }
}
```

```

    }
}

// Select all records
selectButton.setOnClickListener {
    val cursor: Cursor = dbHelper.getAllRecords()
    if (cursor.moveToFirst()) {
        val result = StringBuilder()
        do {
            val id = cursor.getInt(cursor.getColumnIndex(DBHelper.COLUMN_ID))
            val name = cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_NAME))
            val city = cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_CITY))
            result.append("ID: $id, Name: $name, City: $city\n")
        } while (cursor.moveToNext())
        Toast.makeText(this, result.toString(), Toast.LENGTH_LONG).show()
    } else {
        Toast.makeText(this, "No records found", Toast.LENGTH_SHORT).show()
    }
}
}
}
}

```

Activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Name"
        android:minHeight="48dp" />

    <EditText
        android:id="@+id/cityEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter City"
        android:minHeight="48dp" />

    <EditText
        android:id="@+id/idEditText"

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter ID (for update/delete)"
    android:inputType="number"
    android:minHeight="48dp" />
```

```
<Button
    android:id="@+id/insertButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Insert" />
```

```
<Button
    android:id="@+id/updateButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Update" />
```

```
<Button
    android:id="@+id/deleteButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Delete" />
```

```
<Button
    android:id="@+id/selectButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Select All" />
```

```
</LinearLayout>
```