For practical various controls:

MAIN ACTIVITy.kts

```
class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // AutoCompleteTextView items
        val items = arrayOf("Item 1", "Item 2", "Item 3")
        val adapter = ArrayAdapter(this, android.R.layout.simple_dropdown_item_1line, items)
        binding.autoCompleteTextView.setAdapter(adapter)

        binding.buttonSubmit.setOnClickListener {
            // Fetch value from EditText
            val editTextValue = binding.editTextInput.text.toString()

            // Fetch selected RadioButton value
            val selectedRadio = when (binding.radioGroup.checkedRadioButtonId) {
                R.id.radioButton1 -> "Option 1"
                R.id.radioButton2 -> "Option 2"
                else -> "No option selected"
            }

            // Fetch selected CheckBox values
            val selectedCheckBoxes = mutableListOf<String>()
            if (binding.checkBox1.isChecked) selectedCheckBoxes.add("CheckBox 1")
            if (binding.checkBox2.isChecked) selectedCheckBoxes.add("CheckBox 2")

            // Fetch selected AutoComplete value
```

```
        val autoCompleteValue = binding.autoCompleteTextView.text.toString()

        // Combine values and display in TextView
        val result = StringBuilder().apply {
            append("EditText: $editTextValue\n")
            append("RadioButton: $selectedRadio\n")
            append("CheckBoxes: ${selectedCheckBoxes.joinToString(", ")}\n")
            append("AutoComplete: $autoCompleteValue")
        }
        binding.textViewOutput.text = result.toString()
    }
  }
}
```

Practical Make CALCULATOR

```
class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        var input = ""

        binding.button0.setOnClickListener { input += "0" }
        binding.button1.setOnClickListener { input += "1" }
        binding.button2.setOnClickListener { input += "2" }
        binding.button3.setOnClickListener { input += "3" }
        binding.button4.setOnClickListener { input += "4" }
        binding.button5.setOnClickListener { input += "5" }
        binding.button6.setOnClickListener { input += "6" }
```

```
        binding.button7.setOnClickListener { input += "7" }
        binding.button8.setOnClickListener { input += "8" }
        binding.button9.setOnClickListener { input += "9" }
        binding.buttonAdd.setOnClickListener { input += "+" }
        binding.buttonSubtract.setOnClickListener { input += "-" }
        binding.buttonMultiply.setOnClickListener { input += "*" }
        binding.buttonDivide.setOnClickListener { input += "/" }
        binding.buttonClear.setOnClickListener { input = "" }

        binding.buttonEquals.setOnClickListener {
            try {
                val result = eval(input)  // Evaluate the expression
                binding.resultTextView.text = result.toString()
                input = result.toString()  // Reset input to result
            } catch (e: Exception) {
                binding.resultTextView.text = "Error"
            }
        }
    }

    private fun eval(expression: String): Double {
        return expression.toDoubleOrNull() ?: 0.0  // Basic eval (for simplicity, you can extend for
complex expressions)
    }
}


<?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"
    android:padding="16dp">

    <!-- Result Display -->
```

```
<TextView
    android:id="@+id/resultTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="0"
    android:textSize="32sp"
    android:gravity="end"
    android:padding="16dp" />

<!-- Buttons for Calculator -->
<GridLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:columnCount="4"
    android:orientation="horizontal">

    <!-- Row 1 -->
    <Button
        android:id="@+id/button7"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_rowSpan="1"
        android:layout_columnWeight="1"
        android:text="7" />

    <Button
        android:id="@+id/button8"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:text="8" />

    <Button
```

```
    android:id="@+id/button9"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="9" />

<Button
    android:id="@+id/buttonAdd"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="+" />

<!-- Row 2 -->
<Button
    android:id="@+id/button4"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="4" />

<Button
    android:id="@+id/button5"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="5" />

<Button
    android:id="@+id/button6"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="6" />
```

```
<Button
    android:id="@+id/buttonSubtract"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="-" />

<!-- Row 3 -->
<Button
    android:id="@+id/button1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="1" />

<Button
    android:id="@+id/button2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="2" />

<Button
    android:id="@+id/button3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_columnWeight="1"
    android:text="3" />

<Button
    android:id="@+id/buttonMultiply"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
```

```
        android:layout_columnWeight="1"
        android:text="*" />

    <!-- Row 4 -->
    <Button
        android:id="@+id/button0"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="2"
        android:text="0" />

    <Button
        android:id="@+id/buttonClear"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:text="C" />

    <Button
        android:id="@+id/buttonDivide"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_columnWeight="1"
        android:text="/" />

    <!-- Row 5 -->
    <Button
        android:id="@+id/buttonEquals"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="=" />
  </GridLayout>

</LinearLayout>
```

TEMPERATURE CONVERTOR

```
class TemperatureConverterActivity : AppCompatActivity() {

    private lateinit var binding: ActivityTemperatureConverterBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityTemperatureConverterBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.buttonConvert.setOnClickListener {
            val inputTemperature = binding.editTextTemperature.text.toString().toDoubleOrNull()
            if (inputTemperature != null) {
                val result = when {
                    binding.radioCelsiusToFahrenheit.isChecked ->
                        (inputTemperature * 9/5) + 32 // Celsius to Fahrenheit
                    binding.radioFahrenheitToCelsius.isChecked ->
                        (inputTemperature - 32) * 5/9 // Fahrenheit to Celsius
                    else -> null
                }
                binding.textViewResult.text = result?.let { "$it" } ?: "Please select a conversion option."
            } else {
                binding.textViewResult.text = "Please enter a valid temperature."
            }
        }
    }
}
```

```
<?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"
android:padding="16dp"
android:gravity="center">

<!-- Input Field for Temperature -->
<EditText
   android:id="@+id/editTextTemperature"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:hint="Enter Temperature"
   android:inputType="numberDecimal"
   android:textSize="18sp"
   android:padding="16dp" />

<!-- Conversion Type RadioButtons (Celsius to Fahrenheit / Fahrenheit to Celsius) -->
<RadioGroup
   android:id="@+id/radioGroup"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:orientation="horizontal"
   android:padding="16dp">

   <RadioButton
      android:id="@+id/radioCelsiusToFahrenheit"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Celsius to Fahrenheit" />

   <RadioButton
```

```
        android:id="@+id/radioFahrenheitToCelsius"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Fahrenheit to Celsius" />
    </RadioGroup>

    <!-- Convert Button -->
    <Button
        android:id="@+id/buttonConvert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Convert" />

    <!-- Output Result -->
    <TextView
        android:id="@+id/textViewResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Result will appear here"
        android:textSize="24sp"
        android:padding="16dp"
        android:gravity="center" />
</LinearLayout>
```

Animation:

```
package com.example.animation

import android.animation.ObjectAnimator

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import com.example.animation.databinding.ActivityMainBinding
```

```
class MainActivity : AppCompatActivity() {
    private lateinit var binding: ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)



        binding.up.setOnClickListener{
            ObjectAnimator.ofFloat( binding.imageView, "alpha", 0f, 1f).apply {
                duration = 500 // Animation duration: 2 seconds
                start() // Start the animation
            }
        }
        binding.bottom.setOnClickListener{
            ObjectAnimator.ofFloat( binding.imageView, "alpha", 1f, 0f).apply {
                duration = 500 // Animation duration: 2 seconds
                start() // Start the animation
            }
        }
        binding.left.setOnClickListener{
            ObjectAnimator.ofFloat( binding.imageView, "scaleX", 2f, 1f).apply {
                duration = 500 // Animation duration: 2 seconds
                start() // Start the animation
            }
        }
        binding.right.setOnClickListener{


        }
        binding.imageButton.setOnClickListener{
```

```
        ObjectAnimator.ofFloat( binding.imageView, "rotation", 0f, 180f).apply {
            duration = 500 // Animation duration: 2 seconds
            start() // Start the animation
        }
        ObjectAnimator.ofFloat( binding.imageView, "scaleX", 1f, 2f).apply {
            duration = 500 // Animation duration: 2 seconds
            start() // Start the animation
        }
        ObjectAnimator.ofFloat( binding.imageView, "scaleY", 1f, 2f).apply {
            duration = 500 // Animation duration: 2 seconds
            start() // Start the animation
        }
    }


  }
}
```

CALENDAR

1. Setup:

- Ensure you have the `MaterialDatePicker` library in your `build.gradle`:

```
dependencies {
    implementation 'com.google.android.material:material:1.9.0'
}
```

2. Layout File (`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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <Button
```

```
        android:id="@+id/btnSelectDate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Select Date" />

    <TextView
        android:id="@+id/tvSelectedDate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Selected Date will appear here"
        android:gravity="center"
        android:layout_marginTop="16dp"
        android:textSize="16sp" />
</LinearLayout>
```

3. MainActivity Code:

```
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.mycalendarapp.databinding.ActivityMainBinding
import com.google.android.material.datepicker.MaterialDatePicker
import java.text.SimpleDateFormat
import java.util.*

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Set up button click listener
        binding.btnSelectDate.setOnClickListener {
            showDatePicker()
        }
    }

    private fun showDatePicker() {
        // Create Date Picker
        val datePicker = MaterialDatePicker.Builder.datePicker()
            .setTitleText("Select a Date")
            .build()

        // Show Date Picker
        datePicker.show(supportFragmentManager, "DATE_PICKER")

        // Handle Date Selection
        datePicker.addOnPositiveButtonClickListener { selection ->
            // Convert timestamp to readable date
            val dateFormatter = SimpleDateFormat("yyyy-MM-dd",
Locale.getDefault())
            val selectedDate = dateFormatter.format(Date(selection))

            // Set the selected date in the TextView
            binding.tvSelectedDate.text = "Selected Date: $selectedDate"

            // Print the date
```

```
            Toast.makeText(this, "Date Selected: $selectedDate",
Toast.LENGTH_SHORT).show()
        }
    }
}
```

4. Explanation:

- **MaterialDatePicker:** Provides a modern and user-friendly date picker.
- **View Binding:** Simplifies accessing views like `btnSelectDate` and `tvSelectedDate` without `findViewById`.
- **Date Formatting:** Converts the timestamp returned by `MaterialDatePicker` into a human-readable format.

5. Output:

- When the app launches, you will see a button labeled "Select Date."
- Clicking the button opens a date picker. Upon selecting a date, the app will display it in the `TextView` and show a toast message with the selected date.

QUIZ APP

Below is a simple quiz app in Kotlin that doesn't require a database. It uses a hardcoded list of questions and displays them one by one. The app will show the user's score at the end.

---

1. Layout File (`activity_main.xml`)

Create a basic UI for the quiz.

```
<?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"
    android:padding="16dp">

    <TextView
        android:id="@+id/tvQuestion"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Question will appear here"
        android:textSize="18sp"
        android:layout_marginBottom="16dp" />

    <RadioGroup
        android:id="@+id/rgOptions"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```
        <RadioButton
            android:id="@+id/rbOption1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 1" />

        <RadioButton
            android:id="@+id/rbOption2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 2" />

        <RadioButton
            android:id="@+id/rbOption3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 3" />

        <RadioButton
            android:id="@+id/rbOption4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Option 4" />
    </RadioGroup>

    <Button
        android:id="@+id/btnSubmit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Submit"
        android:layout_marginTop="16dp" />
</LinearLayout>
```

---

2. MainActivity Code

Implement the logic for displaying questions and handling answers.

```
import android.os.Bundle
import android.widget.RadioButton
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.quizapp.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding

    // Quiz data
    private val questions = listOf(
        Question(
            "What is the capital of France?",
            listOf("Berlin", "Madrid", "Paris", "Lisbon"),
            2
        ),
        Question(
            "Which is the largest planet in our solar system?",
            listOf("Earth", "Jupiter", "Saturn", "Mars"),
            1
```

```
        ),
        Question(
            "Who wrote 'Hamlet'?",
            listOf("Charles Dickens", "William Shakespeare", "Jane Austen",
"Mark Twain"),
            1
        )
    )

    private var currentQuestionIndex = 0
    private var score = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Display the first question
        showQuestion()

        binding.btnSubmit.setOnClickListener {
            val selectedOptionId = binding.rgOptions.checkedRadioButtonId

            if (selectedOptionId == -1) {
                Toast.makeText(this, "Please select an option",
Toast.LENGTH_SHORT).show()
            } else {
                // Check the answer
                val selectedOption =
findViewById<RadioButton>(selectedOptionId)
                val answerIndex =
binding.rgOptions.indexOfChild(selectedOption)

                if (answerIndex ==
questions[currentQuestionIndex].correctAnswer) {
                    score++
                }

                // Move to the next question or finish the quiz
                currentQuestionIndex++
                if (currentQuestionIndex < questions.size) {
                    showQuestion()
                } else {
                    showScore()
                }
            }
        }
    }

    private fun showQuestion() {
        val question = questions[currentQuestionIndex]
        binding.tvQuestion.text = question.text
        binding.rbOption1.text = question.options[0]
        binding.rbOption2.text = question.options[1]
        binding.rbOption3.text = question.options[2]
        binding.rbOption4.text = question.options[3]
        binding.rgOptions.clearCheck()
    }

    private fun showScore() {
```

```
        Toast.makeText(this, "Quiz Over! Your Score:
$score/${questions.size}", Toast.LENGTH_LONG).show()
        // Reset the quiz
        currentQuestionIndex = 0
        score = 0
        showQuestion()
    }
}

data class Question(
    val text: String,
    val options: List<String>,
    val correctAnswer: Int // Index of the correct answer in the options
list
)
```

How It Works

1. **Question Data**: The questions and answers are hardcoded into a list of `Question` objects.
2. **UI Updates**: The `showQuestion()` function updates the question and options on the screen.
3. **Answer Checking**: When the "Submit" button is clicked, the selected option is checked against the correct answer.
4. **Score Display**: When the quiz ends, the total score is displayed in a `Toast`.

---

3. Run the App

- Each question is displayed one by one.
- The user selects an answer and clicks "Submit."
- At the end of the quiz, the score is shown.

This app is simple and straightforward, making it great for learning or small projects. You can easily extend it by adding more questions, improving UI, or introducing new features like timers or categories.

DIFFERENT LAYOUTS

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView

   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"
android:background="#f0f0f0"
tools:context=".MainActivity">

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <!-- LinearLayout Example -->
    <LinearLayout
        android:id="@+id/linearLayoutSection"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:background="#FFFFFF"
        android:padding="8dp"
        app:layout_constraintTop_toTopOf="parent">

        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Button 1" />

        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Button 2" />
    </LinearLayout>
```

```
<!-- RelativeLayout Example -->
<RelativeLayout
    android:id="@+id/relativeLayoutSection"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_marginTop="16dp"
    android:background="#E0E0E0"
    app:layout_constraintTop_toBottomOf="@id/linearLayoutSection">

    <TextView
        android:id="@+id/centerText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Centered Text"
        android:textSize="18sp" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:text="Bottom Button" />
</RelativeLayout>

<!-- GridLayout Example -->
<GridLayout
    android:id="@+id/gridLayoutSection"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
```

```
    android:columnCount="3"
    android:useDefaultMargins="true"
    android:background="#FFFFFF"
    android:padding="8dp"
    app:layout_constraintTop_toBottomOf="@id/relativeLayoutSection">

    <Button android:layout_columnSpan="1" android:text="Grid 1" />
    <Button android:layout_columnSpan="1" android:text="Grid 2" />
    <Button android:layout_columnSpan="1" android:text="Grid 3" />
    <Button android:layout_columnSpan="1" android:text="Grid 4" />
    <Button android:layout_columnSpan="1" android:text="Grid 5" />
    <Button android:layout_columnSpan="1" android:text="Grid 6" />
</GridLayout>

<!-- CardView Example -->
<com.google.android.material.card.MaterialCardView
    android:id="@+id/cardViewSection"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    app:cardCornerRadius="8dp"
    app:cardElevation="4dp"
    app:layout_constraintTop_toBottomOf="@id/gridLayoutSection">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

        <TextView
            android:layout_width="match_parent"
```

```
            android:layout_height="wrap_content"
            android:text="CardView Example"
            android:textStyle="bold"
            android:textSize="18sp" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="This is a sample card view with some content" />
      </LinearLayout>
    </com.google.android.material.card.MaterialCardView>

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

```
package com.example.myapp

import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.myapp.databinding.ActivityMainBinding // Import generated binding class

class MainActivity : AppCompatActivity() {

  private lateinit var binding: ActivityMainBinding // Declare binding variable

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    // Initialize binding
    binding = ActivityMainBinding.inflate(layoutInflater)
```

```
setContentView(binding.root)

// Set OnClickListeners using binding
binding.button1.setOnClickListener {
    Toast.makeText(this, "Button 1 clicked", Toast.LENGTH_SHORT).show()
}

binding.button2.setOnClickListener {
    Toast.makeText(this, "Button 2 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid1.setOnClickListener {
    Toast.makeText(this, "Grid Button 1 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid2.setOnClickListener {
    Toast.makeText(this, "Grid Button 2 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid3.setOnClickListener {
    Toast.makeText(this, "Grid Button 3 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid4.setOnClickListener {
    Toast.makeText(this, "Grid Button 4 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid5.setOnClickListener {
    Toast.makeText(this, "Grid Button 5 clicked", Toast.LENGTH_SHORT).show()
}

binding.grid6.setOnClickListener {
```

```
        Toast.makeText(this, "Grid Button 6 clicked", Toast.LENGTH_SHORT).show()
    }

    binding.bottomButton.setOnClickListener {
        Toast.makeText(this, "Bottom Button clicked", Toast.LENGTH_SHORT).show()
    }

    binding.centerButton.setOnClickListener {
        Toast.makeText(this, "Center Button clicked", Toast.LENGTH_SHORT).show()
    }
  }
}
```

Below is the updated Kotlin code using **View Binding**. This approach eliminates the need for `findViewById` and provides a more type-safe and efficient way to access views.

MainActivity.kt

```
package com.mc.dbex

import android.database.Cursor
import android.os.Bundle
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.graphics.Insets
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.mc.dbex.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

    private lateinit var binding: ActivityMainBinding
    private lateinit var dbHelper: DBHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Initialize View Binding
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        // Initialize database helper
        dbHelper = DBHelper(this)
```

```
        // Set padding for edge-to-edge UI
        ViewCompat.setOnApplyWindowInsetsListener(binding.main) { view,
insets ->
            val systemBars: Insets =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            view.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom)
            insets
        }

        // Add data to database
        binding.btnAdd.setOnClickListener {
            val name = binding.inputName.text.toString().trim()
            if (name.isNotEmpty()) {
                val isInserted = dbHelper.insertData(name)
                if (isInserted) {
                    Toast.makeText(this, "Data Added",
Toast.LENGTH_SHORT).show()
                    binding.inputName.text.clear() // Clear input
                } else {
                    Toast.makeText(this, "Error Adding Data",
Toast.LENGTH_SHORT).show()
                }
            } else {
                Toast.makeText(this, "Enter a Name",
Toast.LENGTH_SHORT).show()
            }
        }

        // Show data from database
        binding.btnShow.setOnClickListener {
            val cursor: Cursor = dbHelper.getData()
            if (cursor.count == 0) {
                binding.outputText.text = "No data found."
                return@setOnClickListener
            }

            val data = StringBuilder()
            while (cursor.moveToNext()) {
                data.append("ID: ").append(cursor.getInt(0))
                    .append(", Name:
").append(cursor.getString(1)).append("\n")
            }
            binding.outputText.text = data.toString()
        }
    }
}
```

DBHelper.kt

```
package com.mc.dbex

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 {
        private const val DATABASE_NAME = "UserData.db"
        private const val DATABASE_VERSION = 1
        private const val TABLE_NAME = "users"
        private const val COL_ID = "id"
        private const val COL_NAME = "name"
    }

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(
            "CREATE TABLE $TABLE_NAME (" +
                    "$COL_ID INTEGER PRIMARY KEY AUTOINCREMENT, " +
                    "$COL_NAME TEXT)"
        )
    }

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

    fun insertData(name: String): Boolean {
        val db = writableDatabase
        val values = ContentValues().apply {
            put(COL_NAME, name)
        }
        val result = db.insert(TABLE_NAME, null, values)
        return result != -1L // True if insertion was successful
    }

    fun getData(): Cursor {
        val db = readableDatabase
        return db.rawQuery("SELECT * FROM $TABLE_NAME", null)
    }
}
```

Additional Setup

1. **Enable View Binding:** Add the following line in the `build.gradle` file under `android` block:

```
2. buildFeatures {
3.     viewBinding true
4. }
```

5. **XML Layout (`activity_main.xml`):** Your provided XML layout remains unchanged.

This code simplifies accessing views and ensures type safety with **View Binding** while maintaining the functionality of your database application.