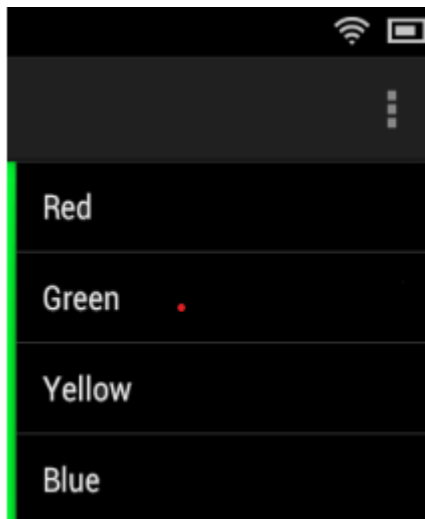


# MOBILE COMPUTING LAB

## Exercise 7:

Create the below activity screen using options Menu.



- Menu item should be displayed from menu.xml file
- When particular menu item is selected, the activity screen background should change to corresponding selected color.
- Use of color.xml for this purpose.
- Also have a Label “Format me” in the Activity. It is with a pop-up menu shows
  - Small
  - Medium
  - Large
  - Change text color

If the user selects any of the first three options, then the “Format me” font size should change accordingly. Use dimen.xml for this purpose. If the user selects fourth option, change the color of the text randomly. (Use color.xml)

In res/menu/menu.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/red"
        android:title="Red" />
    <item
```

```

        android:id="@+id/green"
        android:title="Green" />
    <item
        android:id="@+id/yellow"
        android:title="Yellow" /> <!-- Added Yellow -->
    <item
        android:id="@+id/blue"
        android:title="Blue" />
</menu>

```

In res/menu/popup\_menu.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/small"
        android:title="Small" />
    <item
        android:id="@+id/medium"
        android:title="Medium" />
    <item
        android:id="@+id/large"
        android:title="Large" />
    <item
        android:id="@+id/changeTextColor"
        android:title="Change Text Color" />
</menu>

```

In res/values/dimens.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="text_size_small">30sp</dimen>
    <dimen name="text_size_medium">50sp</dimen>
    <dimen name="text_size_large">70sp</dimen>
</resources>

```

In res/values/colors.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorRed">#FF0000</color>
    <color name="colorGreen">#00FF00</color>
    <color name="colorYellow">#FFFF00</color>
    <color name="colorBlue">#0000FF</color>
    <color name="colorBlack">#000000</color>
    <color name="colorWhite">#FFFFFF</color>

```

```
</resources>
```

In 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">

    <TextView
        android:id="@+id/formatTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Format Me !"
        android:textSize="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
    />

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

In MainActivity.kt :

```
package com.example.myapplication

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.graphics.Color
import android.view.Menu
import android.view.MenuItem
import android.widget.PopupMenu
import android.widget.TextView
import kotlin.random.Random

class MainActivity : AppCompatActivity() {
    private lateinit var formatTextView: TextView
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        formatTextView = findViewById(R.id.formatTextView)
    }
}
```

```

        formatTextView.setOnClickListener {
            showPopupMenu(it)
        }
        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
        }
    }
    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        menuInflater.inflate(R.menu.menu, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        return when (item.itemId) {
            R.id.red -> {
                changeBackgroundColor(R.color.colorRed)
                true
            }
            R.id.green -> {
                changeBackgroundColor(R.color.colorGreen)
                true
            }
            R.id.yellow -> {
                changeBackgroundColor(R.color.colorYellow)
                true
            }
            R.id.blue -> {
                changeBackgroundColor(R.color.colorBlue)
                true
            }
            else -> super.onOptionsItemSelected(item)
        }
    }

    private fun changeBackgroundColor(colorResId: Int) {
        window.decorView.setBackgroundColor(resources.getColor(colorResId,
theme))
    }

    private fun showPopupMenu(view: android.view.View) {
        val popup = PopupMenu(this, view)
        popup.menuInflater.inflate(R.menu.popup_menu, popup.menu)
        popup.setOnMenuItemClickListener { menuItem ->
            when (menuItem.itemId) {
                R.id.small -> {

```

```

        formatTextView.textSize =
resources.getDimension(R.dimen.text_size_small) /
resources.displayMetrics.scaledDensity
        true
    }
    R.id.medium -> {
        formatTextView.textSize =
resources.getDimension(R.dimen.text_size_medium) /
resources.displayMetrics.scaledDensity
        true
    }
    R.id.large -> {
        formatTextView.textSize =
resources.getDimension(R.dimen.text_size_large) /
resources.displayMetrics.scaledDensity
        true
    }
    R.id.changeTextColor -> {
        changeTextColorRandomly()
        true
    }
    else -> false
}
}
popup.show()
}

private fun changeTextColorRandomly() {
    val randomColor = Color.rgb(Random.nextInt(256),
Random.nextInt(256), Random.nextInt(256))
    formatTextView.setTextColor(randomColor)
}
}
}

```

In AndroidManifest.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">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light.DarkActionBar"
        tools:targetApi="31">
        <activity

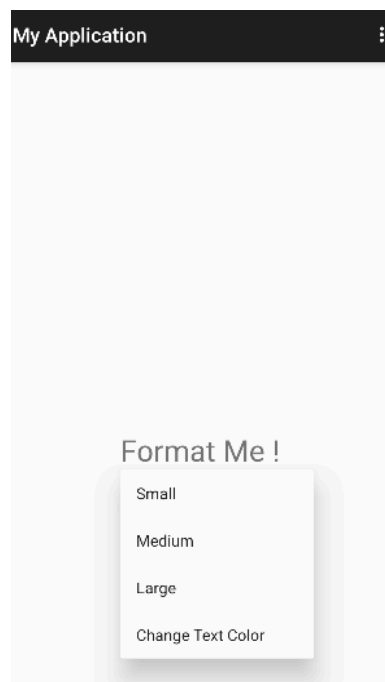
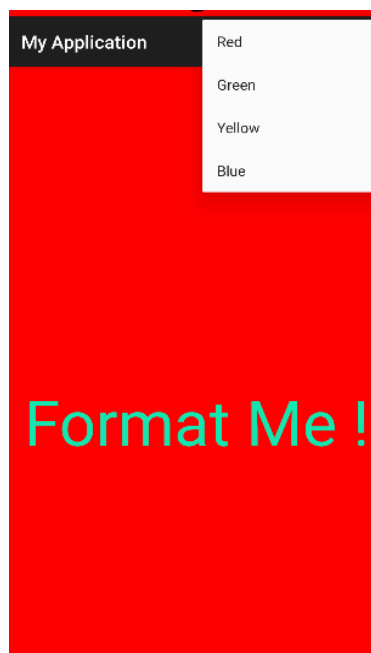
```

```
        android:name=".MainActivity"
        android:exported="true">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

**Output :**



## Exercise 8:

Create a Date difference Android app for the given below screen design using any of the layouts.

The screenshot shows a mobile app interface for calculating the difference between two dates. It features two text input fields at the top: 'From' with the value 'Jan 2, 2018' in orange text, and 'To' with the value 'Feb 7, 2020' in grey text. Below these is a button labeled 'Difference' in orange text, enclosed in a black rectangular border. Underneath the button is a horizontal table with three columns: 'Years', 'Months', and 'Days'. The values '2', '1', and '5' are displayed in the respective columns. The entire interface is set against a light grey background.

Years	Months	Days
2	1	5

When From / To textboxes are clicked, date picker popup and prompt for the date. If the dates are selected in the date picker dialog box, it should be placed in their respective text boxes.

When the difference button is clicked, it should display the difference in the below text view.

(use colors from colors.xml for text colors)

**MainActivity.kt :**

```
package com.example.myapplication

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.app.DatePickerDialog
import android.os.Build
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.annotation.RequiresApi
import java.text.SimpleDateFormat
import java.time.LocalDate
import java.time.Period
import java.util.Calendar
```

```

class MainActivity : AppCompatActivity() {
    private lateinit var fromEditText: EditText
    private lateinit var toEditText: EditText
    private lateinit var yearsTextView: TextView
    private lateinit var monthsTextView: TextView
    private lateinit var daysTextView: TextView
    private lateinit var differenceTextView: TextView
    private var fromDate: Calendar? = null
    private var toDate: Calendar? = null
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        fromEditText = findViewById(R.id.fromEditText)
        toEditText = findViewById(R.id.toEditText)
        val differenceButton: Button = findViewById(R.id.button)
        yearsTextView = findViewById(R.id.yearsTextView)
        monthsTextView = findViewById(R.id.monthsTextView)
        daysTextView = findViewById(R.id.daysTextView)
        differenceTextView = findViewById(R.id.differenceTextView)

        fromEditText.setOnClickListener { showDatePickerDialog(true) }
        toEditText.setOnClickListener { showDatePickerDialog(false) }

        differenceButton.setOnClickListener { calculateDateDifference() }
        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
        }
    }
    private fun showDatePickerDialog(isFromDate: Boolean) {
        val calendar = Calendar.getInstance()
        val datePickerDialog = DatePickerDialog(this,
            { _, year, month, dayOfMonth ->
                val selectedDate = Calendar.getInstance().apply {
                    set(year, month, dayOfMonth)
                }
                val formattedDate = SimpleDateFormat("MMM d,
yyyy").format(selectedDate.time)

                if (isFromDate) {
                    fromDate = selectedDate
                    fromEditText.setText(formattedDate)
                } else {
                    toDate = selectedDate
                    toEditText.setText(formattedDate)
                }
            },
        ),
    },

```



```

        calendar.get(Calendar.YEAR) ,
        calendar.get(Calendar.MONTH) ,
        calendar.get(Calendar.DAY_OF_MONTH)
    )
    datePickerDialog.show()
}

@RequiresApi (Build.VERSION_CODES.O)
private fun calculateDateDifference() {
    if (fromDate != null && toDate != null) {
        val fromLocalDate = LocalDate.of(fromDate!!.get(Calendar.YEAR) ,
fromDate!!.get(Calendar.MONTH) + 1, fromDate!!.get(Calendar.DAY_OF_MONTH) )
        val toLocalDate = LocalDate.of(toDate!!.get(Calendar.YEAR) ,
toDate!!.get(Calendar.MONTH) + 1, toDate!!.get(Calendar.DAY_OF_MONTH) )

        // Calculate the period between two dates
        val period = Period.between(fromLocalDate, toLocalDate)

        yearsTextView.text = "Years \n\n ${period.years}"
        monthsTextView.text = "Months \n\n ${period.months}"
        daysTextView.text = "Days \n\n ${period.days}"
        differenceTextView.text = "Difference calculated successfully."
    } else {
        differenceTextView.text = "Please select both dates."
    }
}
}

```

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"
    android:background="#F8F5F5"
    android:foregroundTint="#FFFFFF"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/fromTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="From"
        android:textColor="@color/text_color_2"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"

```

```
app:layout_constraintHorizontal_bias="0.105"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.223" />
```

<LinearLayout

```
    android:id="@+id/linearLayout"
    android:layout_width="334dp"
    android:layout_height="69dp"
    android:dividerPadding="13dp"
    android:gravity="center|top"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.376"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.628">
```

<TextView

```
    android:id="@+id/yearsTextView"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginEnd="26dp"
    android:text="Years \n"
    android:textAlignment="center"
    android:textColor="@color/text_color_2"
    android:textSize="16sp" />
```

<TextView

```
    android:id="@+id/monthsTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginEnd="26dp"
    android:text="Months \n"
    android:textAlignment="center"
    android:textColor="@color/text_color_2"
    android:textSize="16sp" />
```

<TextView

```
    android:id="@+id/daysTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Days\n"
    android:textAlignment="center"
    android:textColor="@color/text_color_2"
    android:textSize="16sp" />
```

</LinearLayout>

<EditText

```
    android:id="@+id/toEditText"
```

```
    android:layout_width="203dp"
    android:layout_height="48dp"
    android:backgroundTint="#00FFFFFF"
    android:focusable="false"
    android:hint="Select Date"
    android:textColor="@color/text_color_2"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.879"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.31" />
```

#### <View

```
    android:layout_width="334dp"
    android:layout_height="1dp"
    android:background="@color/text_color_2"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.389"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.672" />
```

#### <View

```
    android:layout_width="334dp"
    android:layout_height="1dp"
    android:background="@color/text_color_2"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.389"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.558" />
```

#### <Button

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="#FFFFFFE"
    android:foregroundTint="#FFF7F7"
    android:text="Difference"
    android:textColor="@color/text_color"
    android:textSize="20sp"
    android:textStyle="bold"
    app:cornerRadius="0dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.515"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
```

```
app:layout_constraintVertical_bias="0.443"  
app:strokeColor="#000000"  
app:strokeWidth="3dp" />
```

```
<TextView
```

```
    android:id="@+id/toTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="To"  
    android:textColor="@color/text_color_2"  
    android:textSize="18sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.099"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.309" />
```

```
<EditText
```

```
    android:id="@+id/fromEditText"  
    android:layout_width="202dp"  
    android:layout_height="49dp"  
    android:backgroundTint="#00FFFFFF"  
    android:focusable="false"  
    android:hint="Select Date"  
    android:textColor="@color/text_color"  
    android:textSize="16sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.875"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.218" />
```

```
<TextView
```

```
    android:id="@+id/differenceTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textAlignment="center"  
    android:textColor="@color/text_color"  
    android:textSize="16sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.498"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.732" />
```

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

## Colors.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="text_color">#FF5722</color>
  <color name="background_color">#FFFFFF</color>
  <color name="text_color_2">#73706F</color>
</resources>
```

## Output :

The image displays three sequential screenshots of a date difference calculator application.

**First Screenshot:** The app interface shows two date pickers labeled "From" and "To", each with a "Select Date" button. Below the pickers is a button labeled "Difference". At the bottom, there are input fields for "Years", "Months", and "Days".

**Second Screenshot:** A date picker modal is open, showing the month of January 2025. The date "Wed, Jan 8" is selected and highlighted. The modal includes navigation arrows, a day-of-the-week header, and a grid of dates. "Cancel" and "OK" buttons are at the bottom.

**Third Screenshot:** The app shows the calculated date difference. The "From" date is "Jan 2, 2018" and the "To" date is "Feb 7, 2020". The "Difference" button is still present. Below the date fields, the difference is broken down into "Years: 2", "Months: 1", and "Days: 5". A message at the bottom states "Difference calculated successfully."