activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
   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">

   <!-- AppBar with Toolbar -->
   <com.google.android.material.appbar.AppBarLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content">

       <androidx.appcompat.widget.Toolbar
           android:id="@+id/toolbar"
           android:layout_width="match_parent"
           android:layout_height="?attr/actionBarSize"
           android:background="?attr/colorPrimary"
           android:titleTextColor="@android:color/white"
           app:title="Menu Demo" />
   </com.google.android.material.appbar.AppBarLayout>

   <!-- Your existing layout below the toolbar -->
   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:orientation="vertical"
       android:gravity="center_horizontal"
       android:layout_marginTop="56dp"> <!-- Leave space for Toolbar -->

       <!-- Alert Dialog Button -->
       <Button
           android:id="@+id/buttonShowAlertDialog"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Show Alert Dialog"
           android:layout_marginTop="32dp" />

       <!-- Popup Menu Button -->
       <Button
           android:id="@+id/buttonShowPopupMenu"
           android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:text="Show Popup Menu"
           android:layout_marginTop="32dp" />

       <!-- Context Menu Button -->
       <Button
```

```
            android:id="@+id/buttonShowContextMenu"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Show Context Menu"
            android:layout_marginTop="32dp" />

    </LinearLayout>
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

MainActivity.java

```
package com.example.ad5m2;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.util.Log;
import android.view.ContextMenu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.PopupMenu;
import android.widget.Toast;
import androidx.appcompat.widget.Toolbar;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity {
    Button buttonShowAlertDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        // Button to show AlertDialog
        buttonShowAlertDialog= findViewById(R.id.buttonShowAlertDialog);
        buttonShowAlertDialog.setOnClickListener(v -> showAlertDialog());



        // Button to show PopupMenu
        Button buttonShowPopupMenu = findViewById(R.id.buttonShowPopupMenu);
```

```java
        buttonShowPopupMenu.setOnClickListener(this::showPopupMenu);

        // Button to show ContextMenu
        Button buttonShowContextMenu = findViewById(R.id.buttonShowContextMenu);
        registerForContextMenu(buttonShowContextMenu); // Register the button
for context menu
    }


    // Method to show AlertDialog
    private void showAlertDialog() {
        AlertDialog.Builder builder = new
AlertDialog.Builder(MainActivity.this);
        builder.setTitle("Alert Dialog")
                .setMessage("Do you want to continue?")
                .setPositiveButton("Yes", (dialog, which) -> {
                    // Handle Yes button click
                    Toast.makeText(MainActivity.this, "Yes clicked",
Toast.LENGTH_SHORT).show();
                })
                .setNegativeButton("No", (dialog, which) -> {
                    // Handle No button click
                    Toast.makeText(MainActivity.this, "No clicked",
Toast.LENGTH_SHORT).show();
                })
                .create()
                .show();
    }

    // Method to show OptionDialog
    @Override
    public boolean onCreateOptionsMenu(android.view.Menu menu) {
        getMenuInflater().inflate(R.menu.menu_options, menu);
        return true;
    }

    // Handle options menu item clicks
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();

        if (id == R.id.action_settings) {
            Toast.makeText(this, "Settings clicked", Toast.LENGTH_SHORT).show();
            return true;
        } else if (id == R.id.action_about) {
            Toast.makeText(this, "About clicked", Toast.LENGTH_SHORT).show();
            return true;
        }
        return super.onOptionsItemSelected(item);
    }


    // Method to show PopupMenu
    private void showPopupMenu(View view) {
        // Create the popup menu
        PopupMenu popupMenu = new PopupMenu(MainActivity.this, view);

        // Inflate the menu resource
        getMenuInflater().inflate(R.menu.menu_popup, popupMenu.getMenu());

        // Set listener for menu item clicks
        popupMenu.setOnMenuItemClickListener(item -> {
            // Handle click events for each menu item using if-else
            if (item.getItemId() == R.id.action_edit) {
                Toast.makeText(MainActivity.this, "Edit clicked",
Toast.LENGTH_SHORT).show();
                return true; // Indicates the event was handled
            } else if (item.getItemId() == R.id.action_delete) {
                Toast.makeText(MainActivity.this, "Delete clicked",
Toast.LENGTH_SHORT).show();
                return true; // Indicates the event was handled
            } else {
                return false; // Default case to handle unknown items
            }
        });

        // Show the popup menu
        popupMenu.show();
    }

    // Method to create context menu
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        getMenuInflater().inflate(R.menu.menu_context, menu);
    }

    // Handle context menu item selection
    @Override
    public boolean onContextItemSelected(MenuItem item) {
        // Handle item selection using if-else
        if (item.getItemId() == R.id.action_copy) {
            Toast.makeText(this, "Copy clicked", Toast.LENGTH_SHORT).show();
            return true; // Indicates the event was handled
        } else if (item.getItemId() == R.id.action_paste) {
            Toast.makeText(this, "Paste clicked", Toast.LENGTH_SHORT).show();
            return true; // Indicates the event was handled
        } else {
            return super.onContextItemSelected(item); // Default case calls the
parent method
        }
    }

}
```

Res -> menu -> menu_context.xml

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

</menu>
```

Res -> menu -> menu_popup.xml

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

Res -> menu -> menu_options.xm

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