***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">  
  
 <EditText  
 android:id="@+id/popupmenu"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="btnPopup"  
 android:hint="Select Item"  
 android:editable="false"  
 android:gravity="center"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

***MainActivity.java***

package com.example.menu;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.PopupMenu;  
import android.widget.Toast;  
  
import java.util.zip.Inflater;  
  
public class MainActivity extends AppCompatActivity {  
  
 Menu myMenu;  
 EditText selectItem;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 }  
   
 //This function is to create option menu  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.*main\_menu*, menu);  
  
 selectItem = (EditText)findViewById(R.id.*popupmenu*);  
  
 return true;  
 }  
  
 //This function is for making the option menu clickable  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 if (item.getItemId() == R.id.*item1*)  
 {  
 Toast.*makeText*(this, "Home Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
 if (item.getItemId() == R.id.*item2*)  
 {  
 Toast.*makeText*(this, "About Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
 if (item.getItemId() == R.id.*item3*)  
 {  
 Toast.*makeText*(this, "Contact Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 return true;  
 }  
  
 //This is for option menu, this function will trigger once the gadget against this  
 //function is clicked.  
 public void btnPopup(View view) {  
 PopupMenu popup = new PopupMenu(this, view);  
 popup.getMenuInflater().inflate(R.menu.*main\_menu*, popup.getMenu());  
 popup.show();  
   
  
 //This function is to make the popup menu clickable  
 popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem menuItem) {  
 if (menuItem.getItemId() == R.id.*item1*)  
 {  
 selectItem.setText("Home");  
 Toast.*makeText*(MainActivity.this, "Home Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
 if (menuItem.getItemId() == R.id.*item2*)  
 {  
 selectItem.setText("About");  
 Toast.*makeText*(MainActivity.this, "About Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
 if (menuItem.getItemId() == R.id.*item3*)  
 {  
 selectItem.setText("Contact");  
 Toast.*makeText*(MainActivity.this, "Contact Selected", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 return true;  
 }  
 });  
 }  
}

***Colors.xml***

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
 <color name="colorPrimary">#6200EE</color>  
 <color name="colorPrimaryDark">#3700B3</color>  
 <color name="colorAccent">#03DAC5</color>  
</resources>

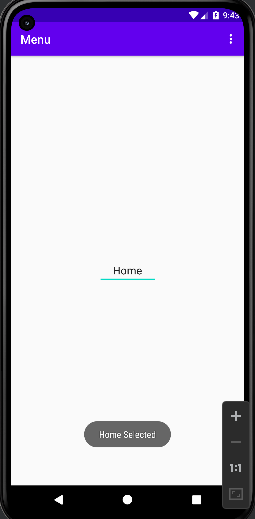
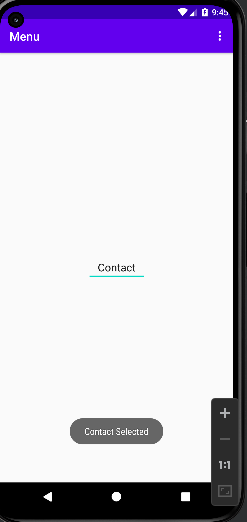
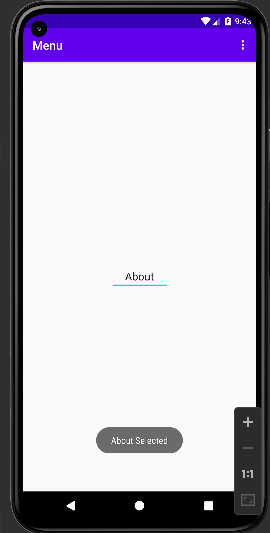
***Strings.xml***

<resources>  
 <string name="app\_name">Menu</string>  
</resources>

***Styles.xml***

<resources>  
 <!-- Base application theme. -->  
 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">  
 <!-- Customize your theme here. -->  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
 </style>  
  
</resources>

**Output:**

** **