**Sanika Kadam 2124UCSF1112**

**Long Click**

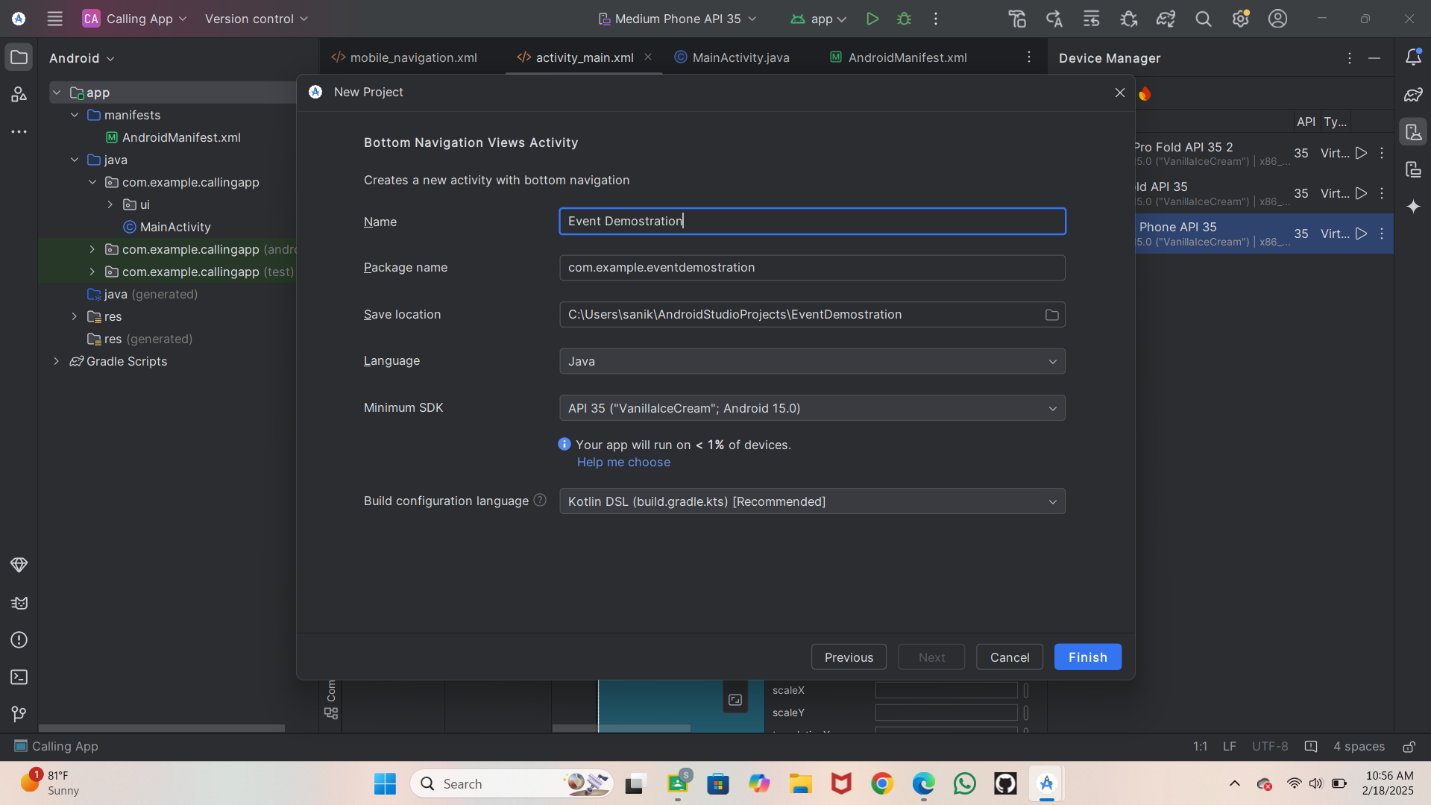
**1. Long Click**

**2. Tools & Technologies Used**

* Android Studio
* Java/Kotlin (Java)
* Emulator or Physical Device

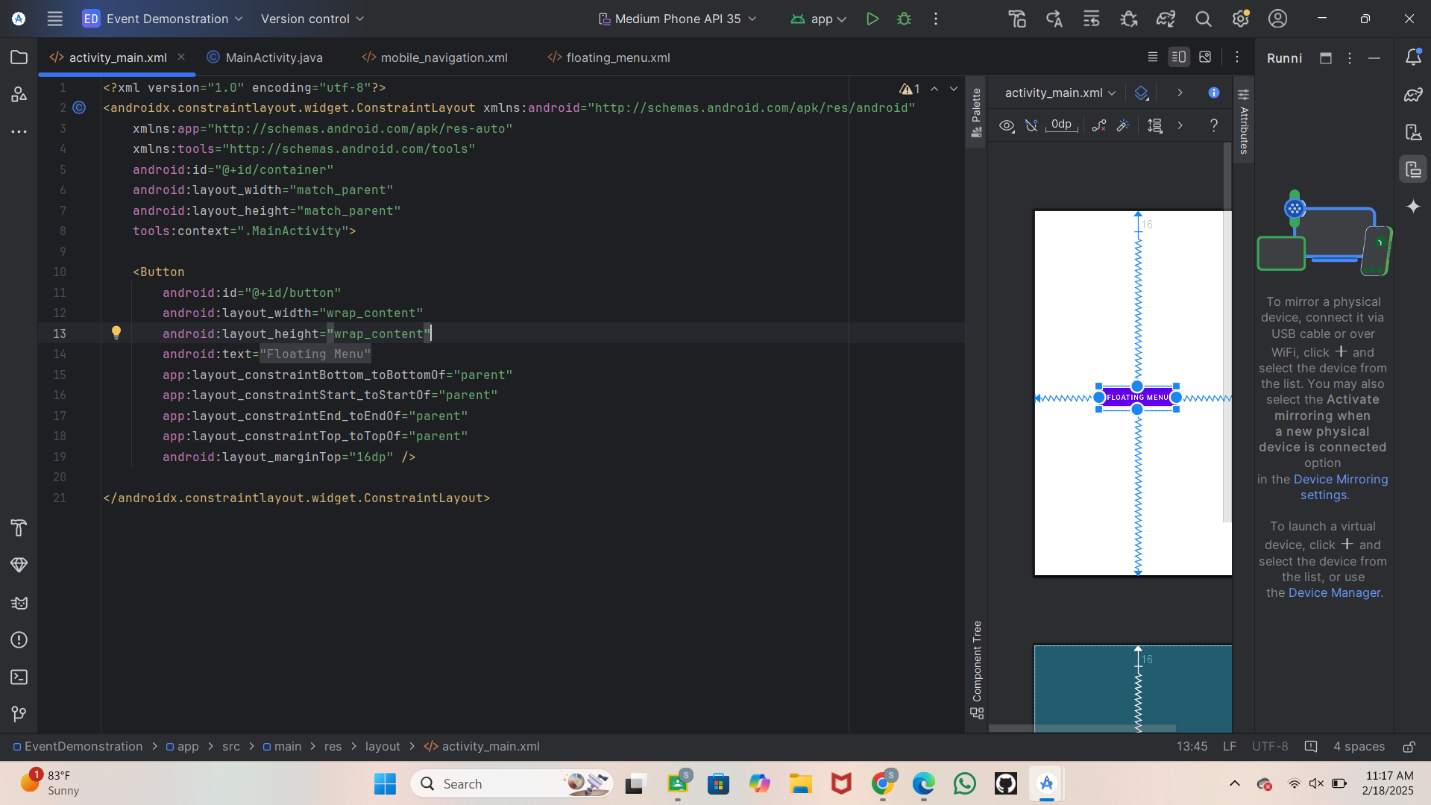
**3. Procedure & Steps**

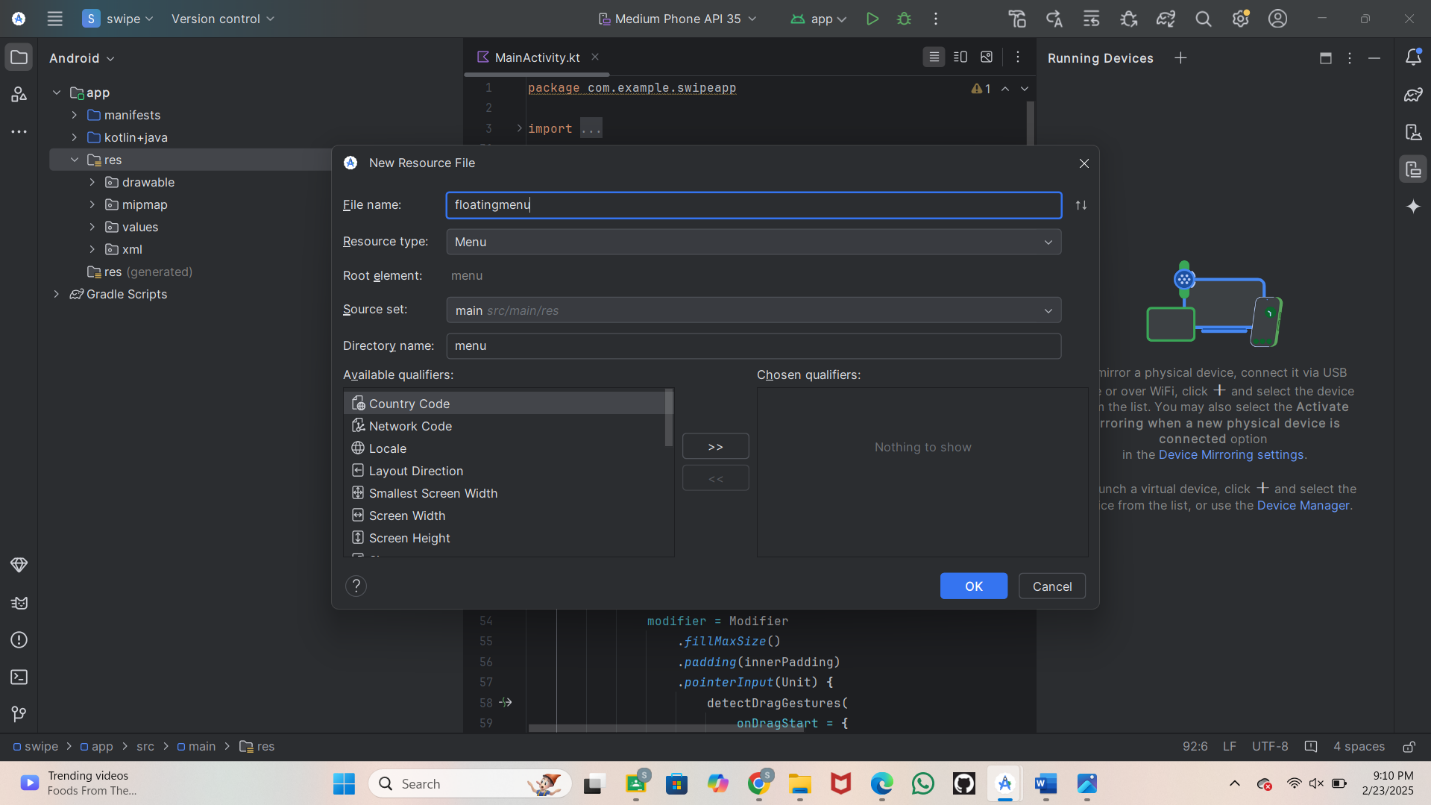
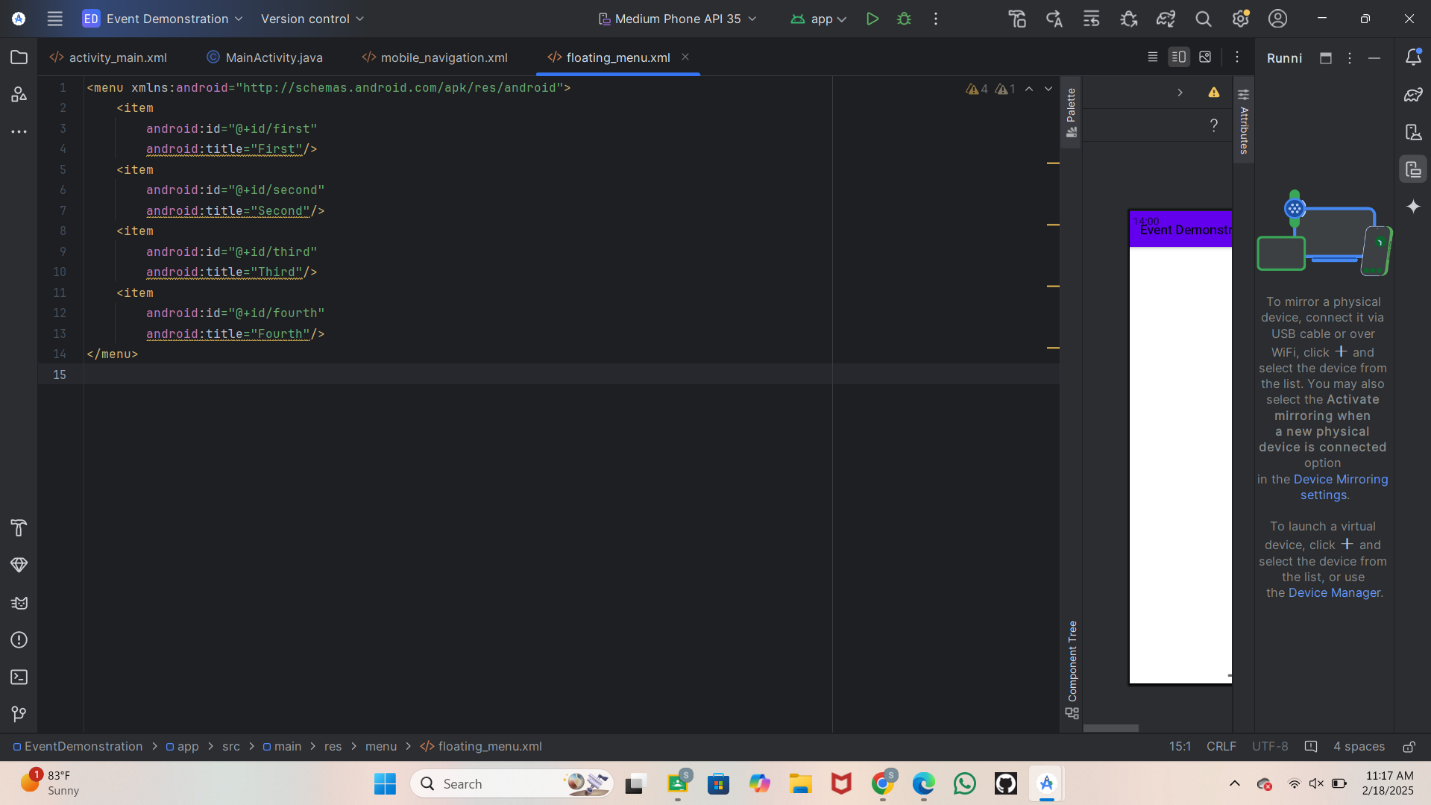
**Step 1: Create a New Project**

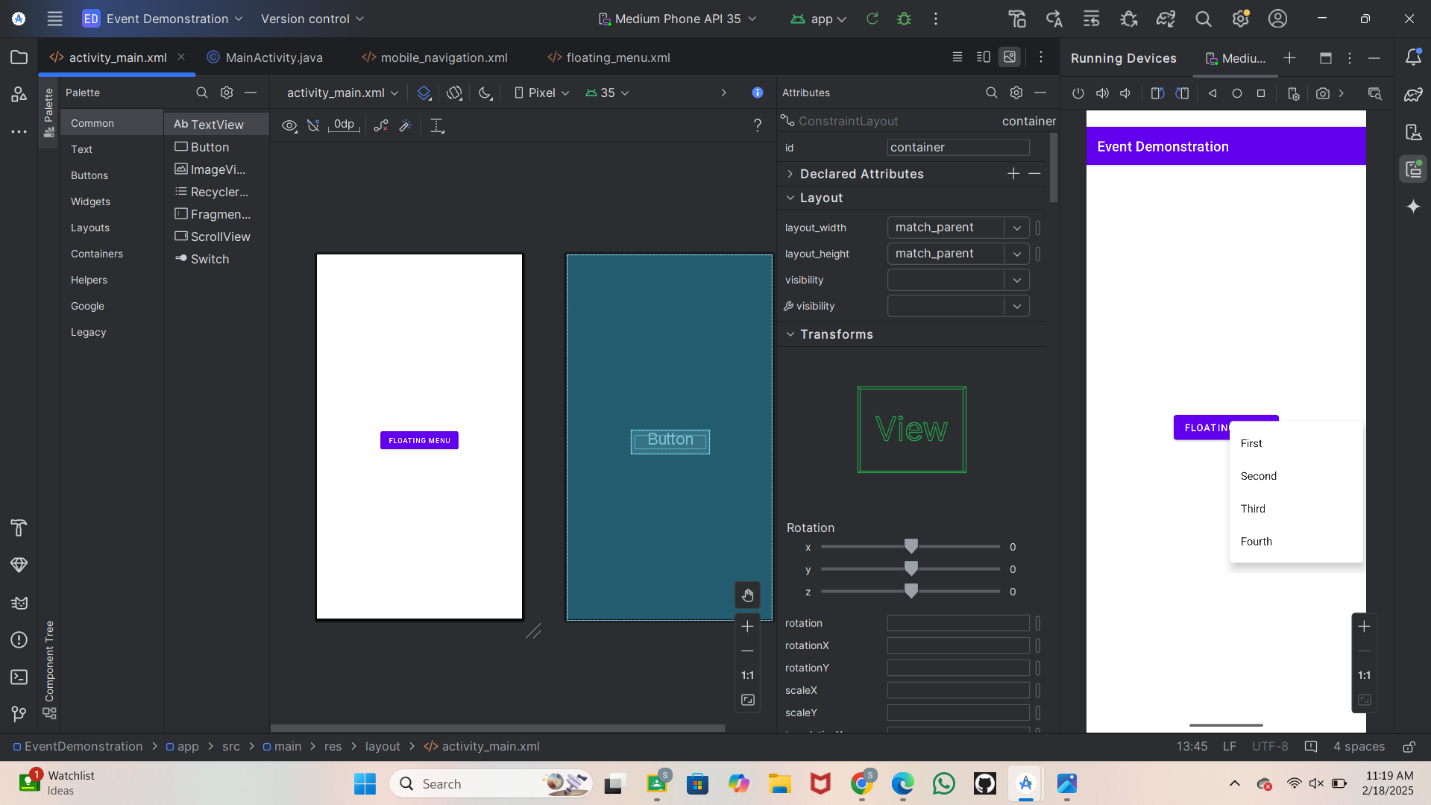
* Open Android Studio and create a new project.
* Select Bottom Navigation Views Activity
* Set the project name and package name.
* Select the programming language (Java).
* Screenshot 

**Step 2: Designing the UI**

* Open activity\_main.xml and design the layout using XML.
* On Left side go on a java then com.example, floatingcontextmenu
* Then take main activity java
* Write a code in activity main xml
* ScreenShot :



* **Step 3**
* Write a code in main activity java
* Go on res then new then Take android resource file
* Screenshot
* **Step 4**
* Write a code in floating menu
* Screenshot
* 
* Run the application



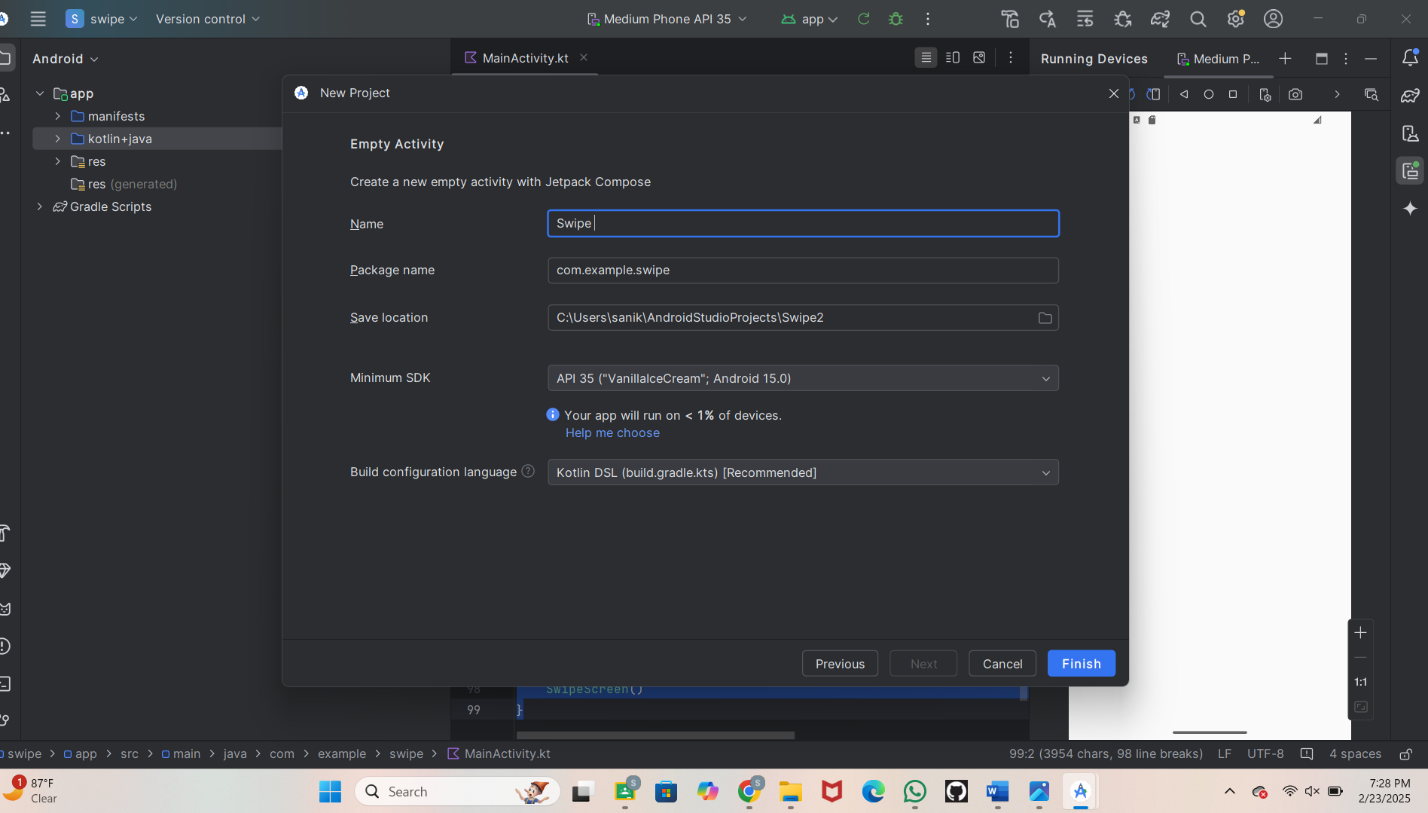
**Swipe 1. Tools & Technologies Used**

* Android Studio
* Java/Kotlin (Kotlin)
* Emulator or Physical Device

**2. Procedure & Steps**

**Step 1: Create a New Project**

* Open Android Studio and create a new project.
* Select Empty Views Activity
* Set the project name and package name.
* Select the programming language (Kotlin).

Screenshot 

**Step 4**

* Write a Code in Main Activity kt

package com.example.swipeapp

import android.os.Bundle

import android.util.Log

import android.widget.Toast

import androidx.activity.ComponentActivity

import androidx.activity.compose.setContent

import androidx.compose.foundation.gestures.detectDragGestures

import androidx.compose.foundation.layout.Box

import androidx.compose.foundation.layout.fillMaxSize

import androidx.compose.foundation.layout.padding

import androidx.compose.material3.MaterialTheme

import androidx.compose.material3.Scaffold

import androidx.compose.material3.SnackbarDuration

import androidx.compose.material3.SnackbarHost

import androidx.compose.material3.SnackbarHostState

import androidx.compose.material3.Text

import androidx.compose.runtime.Composable

import androidx.compose.runtime.getValue

import androidx.compose.runtime.mutableStateOf

import androidx.compose.runtime.remember

import androidx.compose.runtime.rememberCoroutineScope

import androidx.compose.runtime.setValue

import androidx.compose.ui.Alignment

import androidx.compose.ui.Modifier

import androidx.compose.ui.input.pointer.pointerInput

import androidx.compose.ui.tooling.preview.Preview

import androidx.compose.ui.platform.LocalContext

import kotlinx.coroutines.launch

import kotlin.math.absoluteValue

class MainActivity : ComponentActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContent {

SwipeScreen()

}

}

}

@Composable

fun SwipeScreen() {

val snackbarHostState = remember { SnackbarHostState() }

val scope = rememberCoroutineScope()

var swipeDirection by remember { mutableStateOf<String?>(null) }

val context = LocalContext.current // Access context for Toast

MaterialTheme {

Scaffold(

modifier = Modifier.fillMaxSize(),

snackbarHost = { SnackbarHost(snackbarHostState) }

) { innerPadding ->

Box(

modifier = Modifier

.fillMaxSize()

.padding(innerPadding)

.pointerInput(Unit) {

detectDragGestures(

onDragStart = {

swipeDirection = null

},

onDragEnd = {

swipeDirection?.let { direction ->

Log.d("SwipeEvent", "$direction Detected")

scope.launch {

snackbarHostState.showSnackbar(

message = "$direction Detected",

duration = SnackbarDuration.Short

)

}

// Show Toast on swipe detection

Toast.makeText(context, "$direction Detected", Toast.LENGTH\_SHORT).show()

}

},

onDrag = { change, dragAmount ->

change.consume()

if (swipeDirection == null) {

swipeDirection = if (dragAmount.x.absoluteValue > dragAmount.y.absoluteValue) {

if (dragAmount.x > 0) "Swipe Right" else "Swipe Left"

} else {

if (dragAmount.y > 0) "Swipe Down" else "Swipe Up"

}

}

}

)

},

contentAlignment = Alignment.Center

) {

Text(text = "Swipe in any direction")

}

}

}

}

@Preview(showBackground = true)

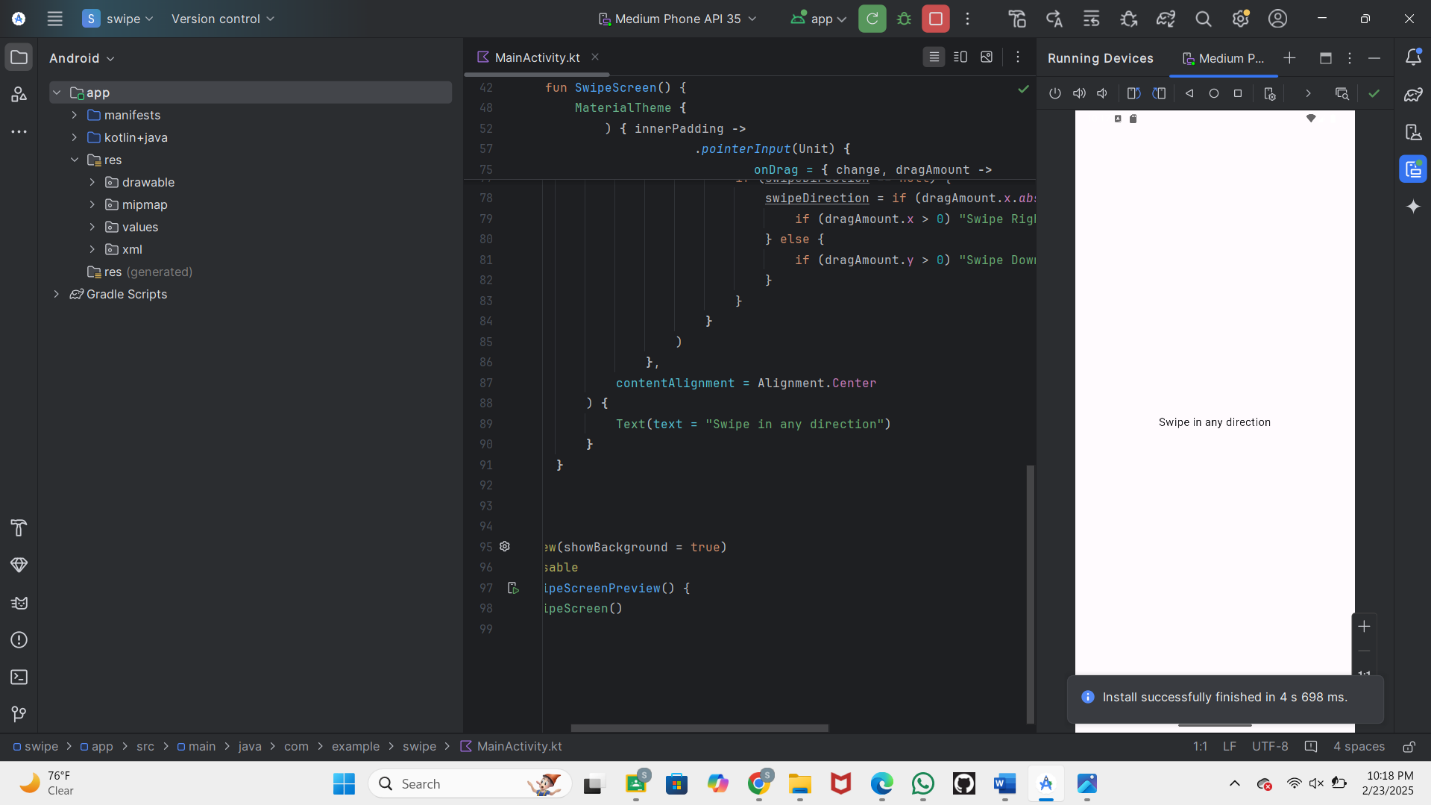
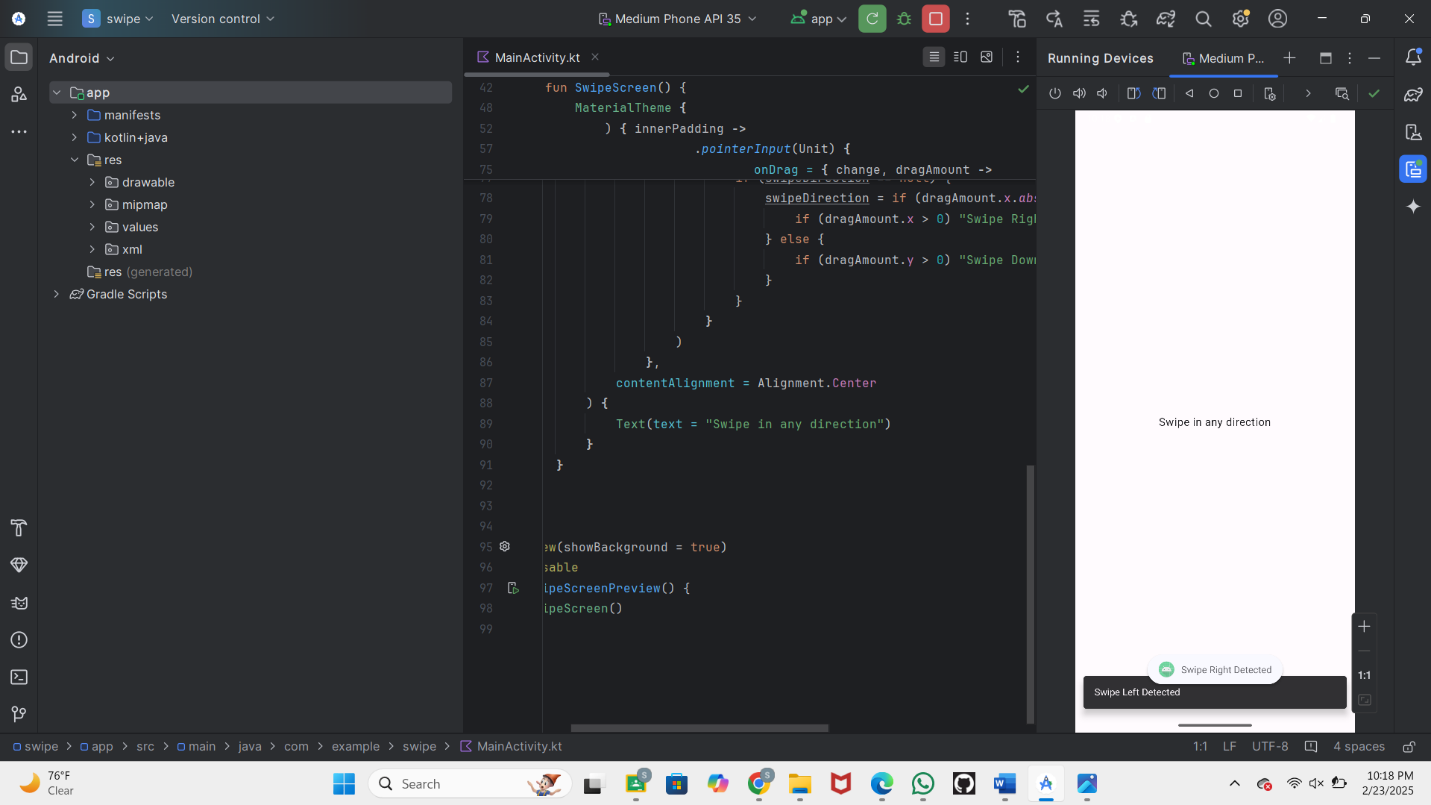
@Composable

fun SwipeScreenPreview() {

SwipeScreen()

}

**Step 5**

* Run the Application
* Screenshot
* 
* Swipe 

**Button Click**

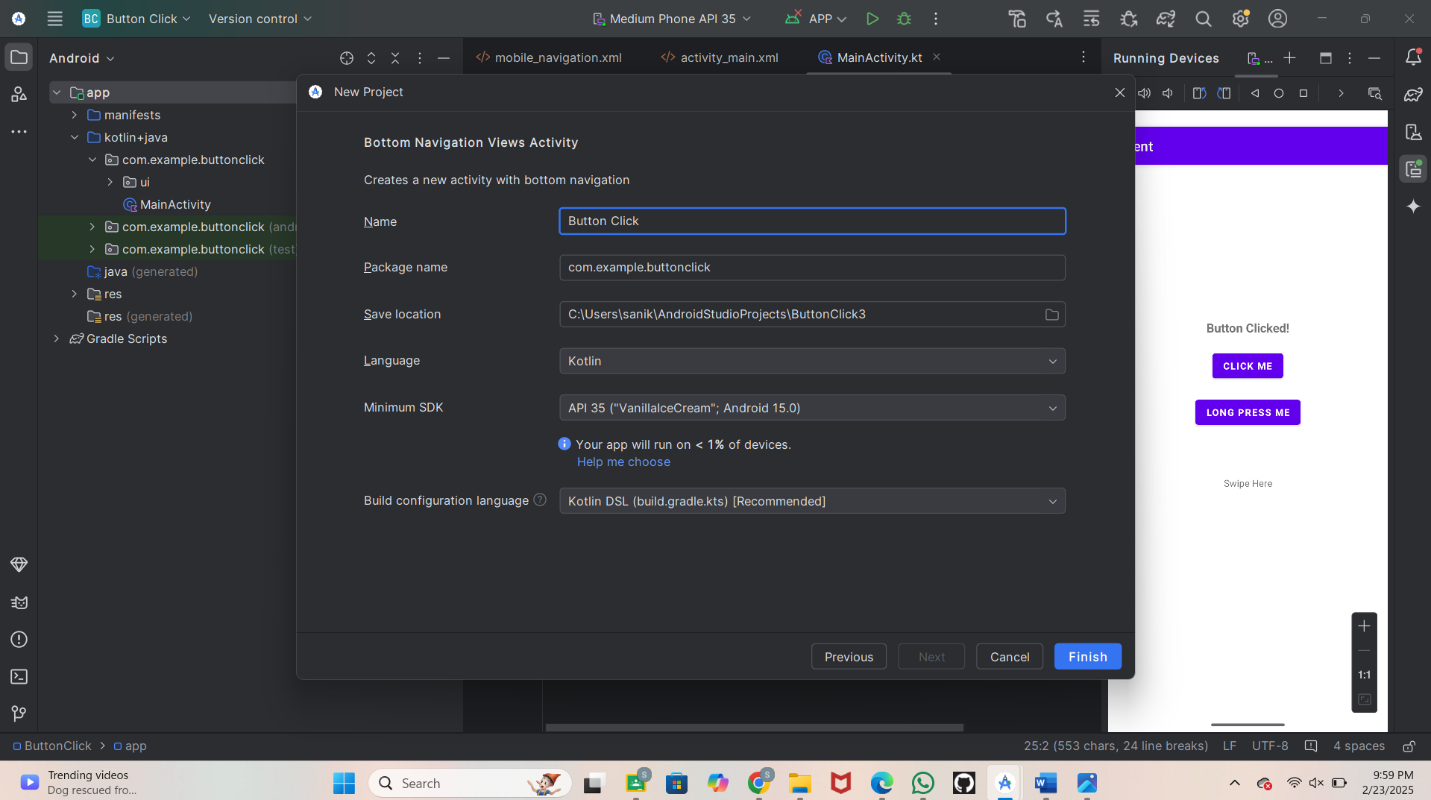
**Step 1: 2. Tools & Technologies Used**

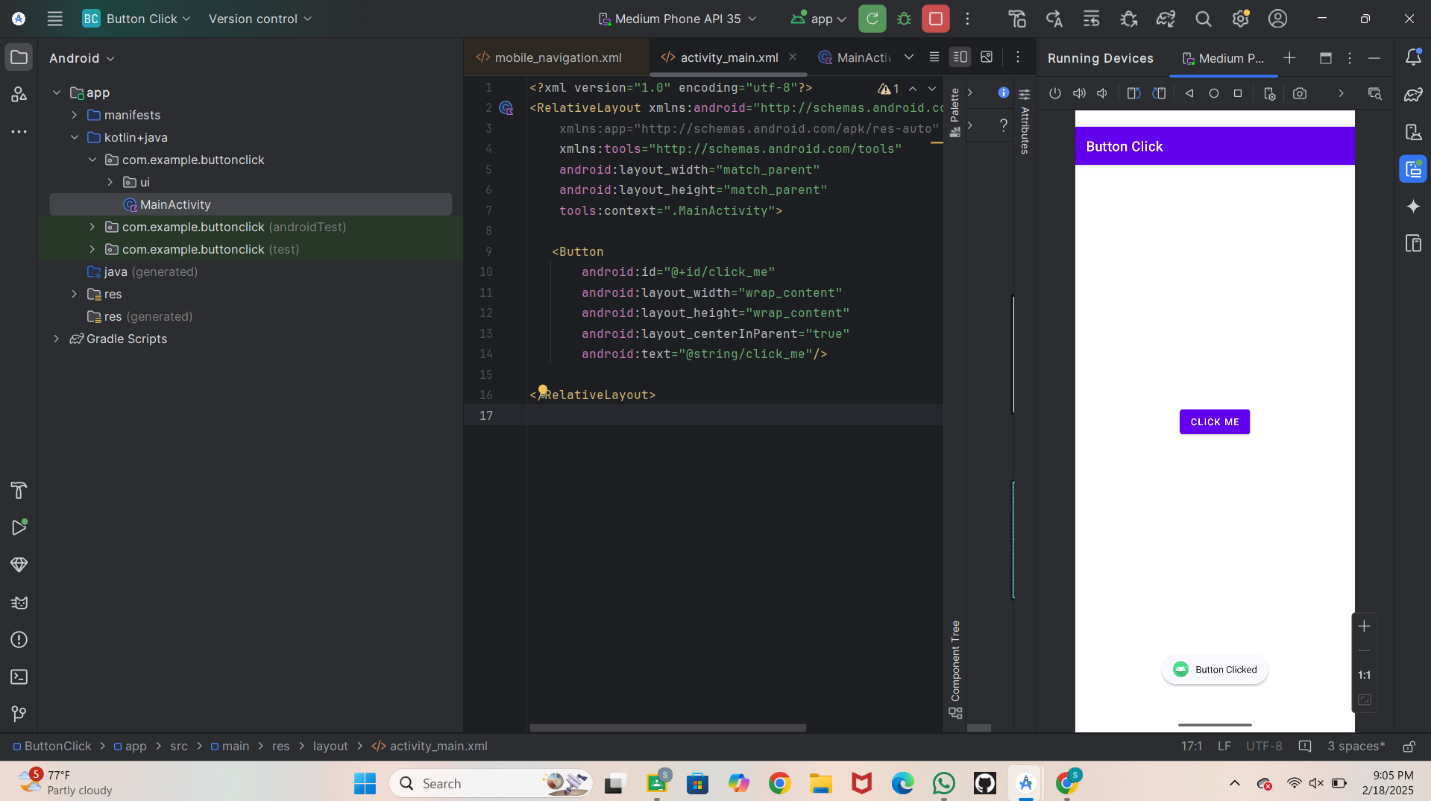
* Android Studio
* Java/Kotlin (Java)
* Emulator or Physical Device

**3. Procedure & Steps**

**Step 1: Create a New Project**

* Open Android Studio and create a new project.
* Select Bottom Navigation Views Activity
* Set the project name and package name.
* Select the programming language (Java).

Screenshot

* **Step 3**
* Write a code in activity main xml
* ?xml version="1.0" encoding="utf-8"?>  
  <RelativeLayout 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">  
    
   <Button  
   android:id="@+id/click\_me"  
   android:layout\_width="wap\_content"  
   android:layout\_height="wrap\_content"  
   android:layout\_centerInParent="true"  
   android:text="@string/click\_me"/>  
    
  </RelativeLayout>
* **Step 4**
* On Left side go on a java then com.example, floatingcontextmenu
* Then take main activity kotin
* Write a code in Main activity kt
* package com.example.buttonclick  
    
  import android.os.Bundle  
  import android.widget.Button  
  import android.widget.Toast  
  import androidx.appcompat.app.AppCompatActivity  
    
  class MainActivity : AppCompatActivity() {  
    
    
    
   override fun onCreate(savedInstanceState: Bundle?) {  
   super.onCreate(savedInstanceState)  
   setContentView(R.layout.*activity\_main*)  
    
   val clickme = findViewById<Button>(R.id.*click\_me*)  
    
   clickme.setOnClickListener **{** Toast.makeText( this, "Button Clicked" ,Toast.*LENGTH\_SHORT*).show()  
    
   **}** }  
  }
* Step 5
* Run the application
* Screenshot
* 
* **Conclusion**
* This program demonstrates how to handle different types of user interaction events in an Android application. the app responds to button clicks, long presses, and swipes.