Ex. No. 10	TabLayout with Swipeable Views using ViewPager2
Date of Exercise	30.10.2024

Aim

To write a kotlin program to tab layout with swipeable views using viewpager2.

Description

TabLayout with Swipeable Views Using ViewPager2 refers to a user interface design pattern in Android development that allows users to navigate between different fragments or views by swiping horizontally. This setup typically involves combining a TabLayout with ViewPager2 to create a smooth and interactive way for users to switch between different tabs or pages within an application.

Key Components:

1. TabLayout: This is a UI element from the Android Material Design library that displays a set of tabs. Each tab can represent a different section of the app, and users can click on a tab to jump directly to that section.
2. ViewPager2: An improved version of the original ViewPager, ViewPager2 allows users to swipe left or right to navigate between fragments or views. It provides better performance and more flexibility, including support for vertical scrolling and right-to-left layouts.

1. Add dependencies to the app-level build
2. Add a TabLayout and ViewPager2 to the activity layout
3. Create fragments for each tab
4. Create a FragmentStateAdapter to manage the fragments in the ViewPager2
5. Link the TabLayout to the ViewPager using setupWithViewPager()

Program:

MainActivity:

```
package com.example.ex_10

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.viewpager2.widget.ViewPager2
import com.example.ex_10.R // Ensure this import is correct
 // Ensure this import matches your package name
import com.example.exp_10.ViewPagerAdapter
import com.google.android.material.tabs.TabLayout
import com.google.android.material.tabs.TabLayoutMediator

class MainActivity : AppCompatActivity() {
    private lateinit var tabLayout: TabLayout
    private lateinit var viewPager: ViewPager2
    private lateinit var viewPagerAdapter: ViewPagerAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main) // Make sure this matches your layout file name

        tabLayout = findViewById(R.id.tabLayout)
        viewPager = findViewById(R.id.viewPager)

        viewPagerAdapter = ViewPagerAdapter(this) // Ensure ViewPagerAdapter is correctly implemented
        viewPager.adapter = viewPagerAdapter

        // Setting up the TabLayout with the ViewPager2
        TabLayoutMediator(tabLayout, viewPager) { tab, position ->
            tab.text = when (position) {
                0 -> "Products"
                1 -> "Shop Details"
```

```
            2 -> "Offers"
            else -> null
        }
    }.attach()
  }
}
```

OfferAdapter.kt:

```
package com.example.exp_10

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class OffersAdapter(private val offers: List<String>) :
RecyclerView.Adapter<OffersAdapter.OfferViewHolder>() {

  class OfferViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
    val offerName: TextView = itemView.findViewById(android.R.id.text1)
  }

  override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): OfferViewHolder {
    val view = LayoutInflater.from(parent.context).inflate(android.R.layout.simple_list_item_1, parent, false)
    return OfferViewHolder(view)
  }

  override fun onBindViewHolder(holder: OfferViewHolder, position: Int) {
    holder.offerName.text = offers[position]
  }

  override fun getItemCount(): Int = offers.size
}
```

Output:

Result:

Thus the code is executed successfully and the output is displayed in the console window.