PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043

Department of Electronics & Telecommunication Engineering

BATCH AND ROLL NO:
EXPERIMENT NO.7
TITLE: Design a mobile application using image slider to show images.
DATE OF PERFORMANCE:
DATE OF SUBMISSION:

Title: Design a mobile application using image slider to show images.

Requirements:
1 Android studio
2. Image Slider Library

Theory:

Introduction

In the vibrant realm of mobile application development, creating visually engaging and interactive user interfaces is paramount. One impactful way to showcase a collection of images seamlessly is through the use of an image slider. This lab focuses on designing a mobile application that incorporates an image slider, allowing users to effortlessly navigate through a series of images, fostering a dynamic and visually appealing user experience.

Objective of the Lab: The primary objective of this lab is to guide you through the process of designing a mobile application that integrates an image slider component. By the end of this lab, you should be proficient in implementing and customizing an image slider, providing users with an immersive and visually captivating way to view a series of images within the application.

Components of the Application:

1. **Image Slider:**
    - An image slider, often referred to as a carousel, is a UI component that allows users to scroll through a set of images horizontally.
    - It enhances the presentation of visual content, making it suitable for showcasing images, banners, or other graphic elements.

Lab Prerequisites:

- Basic understanding of mobile application development concepts.
- Familiarity with the chosen development environment (e.g., Android Studio).
- Prior knowledge of programming languages such as Java (for Android).

Steps:

Step 1: Set Up Your Development Environment

- Ensure that you have Android Studio installed and configured on your machine.

Step 2: Create a New Project

- Open Android Studio and create a new project.
- Choose an appropriate project template, such as "Empty Activity" or "Basic Activity."

Step 3: Design the Main Activity Layout

- Open the XML layout file associated with your main activity (e.g., activity_main.xml).
- Add a ViewPager element to your layout. The ViewPager will serve as the container for the image slider.

Step 4: Implement the Java Code

- Open the Java or Kotlin file associated with your main activity (e.g., MainActivity.java).
- Implement the logic for creating and populating the ViewPager with images.

Step 5: Create an ImageSliderAdapter

- Create an adapter class (e.g., ImageSliderAdapter) to handle the image slider's content.

Step 6: Design Item Layout for ViewPager

- Create a layout file (e.g., item_image.xml) to define the appearance of each item in the ViewPager.

Step 7: Test Your Application

- Run your application on an emulator or a physical device.
- Verify that the image slider displays the specified images.

XML Code:

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">
<com.denzcoskun.imageslider.ImageSlider
android:layout_width="match_parent"
android:layout_height="match_parent"
android:id="@+id/imageslider" app:iss_auto_cycle="true"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

Java Code:

```
package com.example.exp7; import
androidx.appcompat.app.AppCompatActivity; import
android.os.Bundle; import
com.denzcoskun.imageslider.ImageSlider; import
com.denzcoskun.imageslider.constants.ScaleTypes; import
com.denzcoskun.imageslider.models.SlideModel; import
java.util.ArrayList; import java.util.List; public class
MainActivity extends AppCompatActivity {
@Override protected void onCreate(Bundle
savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

```
ImageSlider imageSlider=findViewById(R.id.imageslider); List<SlideModel>
slideModelList; slideModelList = new ArrayList<>(); slideModelList.add(new
SlideModel(R.drawable.sample1,"1", ScaleTypes.FIT)); slideModelList.add(new
SlideModel(R.drawable.sample2,"2",ScaleTypes.FIT));
imageSlider.setImageList(slideModelList);
}}
```

Output:

Conclusion:

……

……

……