



<b>BATCH AND ROLL NO: P5-42114</b>
<b>EXPERIMENT NO.6</b>
<b>TITLE:</b> Design a mobile application to Show any website using web view.
<b>DATE OF PERFORMANCE:</b>
<b>DATE OF SUBMISSION:</b>

**Title:** Design a mobile application to Show any website using web view.

**Requirements:**

1 Android studio

**Theory:**

**Introduction**

The integration of web content into mobile applications has become an integral aspect of enhancing user experiences. In this lab, we will focus on designing a mobile application that incorporates a WebView component. The WebView allows the seamless display of web content within the application, providing users with the convenience of accessing external websites without leaving the app environment.

**Objective of the Lab:** The primary objective of this lab is to guide you through the process of designing a mobile application that utilizes a WebView to showcase content from external websites. By the end of this lab, you should be adept at implementing and customizing the WebView component, offering users a cohesive experience as they navigate web content within the confines of your mobile application.

**Components of the Application:**

**1. WebView Component:**

- The WebView is a versatile component that allows the embedding of web content directly into a mobile application.
- It enables users to interact with and view external websites seamlessly, enhancing the overall application functionality.

**Lab Prerequisites:**

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



**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 WebView component to your layout. You can use the WebView element in your XML file.

**Step 4: Implement the Java Code**

- Open the Java file associated with your main activity (e.g., MainActivity.java).
- In the onCreate method, retrieve the reference to the WebView from the XML layout using findViewById.
- Configure the WebView settings, such as enabling JavaScript, if required.

**Step 5: Test Your Application**

- Run your application on an emulator or a physical device.
- Verify that the WebView displays the content from the specified website.

**Step 6: Enhance WebView Functionality (Optional)**

- Implement additional features, such as handling WebView events (e.g., page loading), managing navigation, or enabling WebView controls.



**XML Code:**

```
<?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">

    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/webViewId"/>
</RelativeLayout>
```

**Java Code:**

```
package com.example.web_viewer_expt6;

import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;

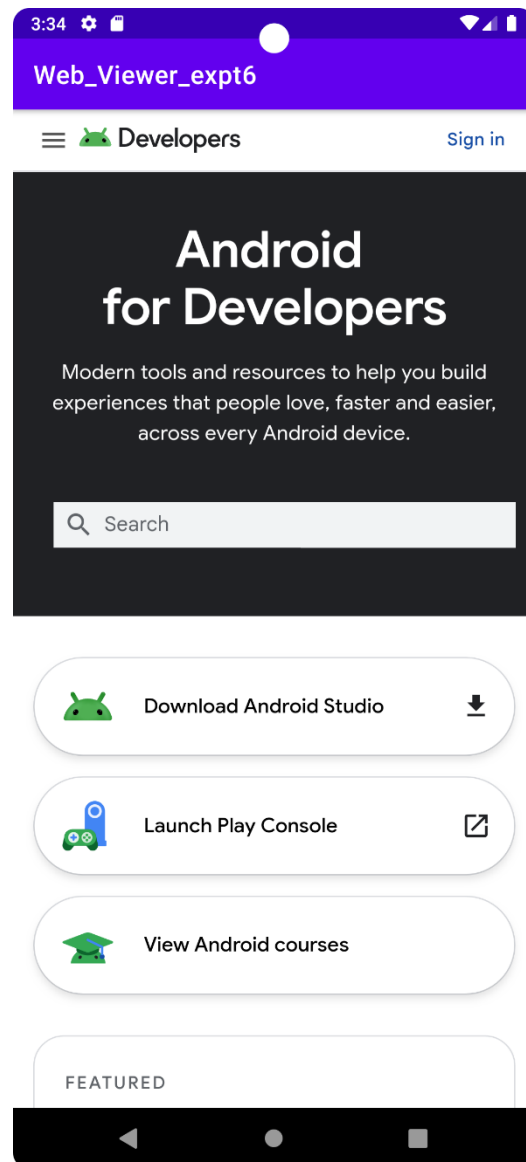
public class MainActivity extends AppCompatActivity {
    WebView webView;
    @SuppressLint("SetJavaScriptEnabled")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        webView=findViewById(R.id.webViewId);
        webView.setWebViewClient(new WebViewClient());

        WebSettings webSettings=webView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        webView.loadUrl("https://developer.android.com");
    }
}
```



**Output:**



**Conclusion:**

.....

.....

.....