SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

UCS1711 - MOBILE APPLICATION DEVELOPMENT LAB

Assignment 12

Name: Jayannthan P T *Dept: CSE 'A'* *Roll No.: 205001049*

App to display a Web Page

Ex. No:12 **Date:**5/9/2023

Title of the Program: Develop an android application to display a static web page with contents that is constructed using formatting tags. Also, should load the web page if present in the specified URL.

Objective:

The objective of the WebPageDisplay Android App is to create an application that can display both a static HTML page constructed with formatting tags and load a web page from a specified URL. The app includes a WebView to render HTML content and allows users to input a URL to load a dynamic webpage.

Algorithm:

1. Create an Android app with two main components: MainActivity.java and activity_main.xml.
2. Initialize WebView, EditText, and Button in MainActivity.java.
3. Enable JavaScript in WebView settings for enhanced functionality.
4. Create a static HTML string with formatting tags to display a stylized page.
5. Use loadData method to load the static HTML content into the WebView.
6. Set up a WebViewClient to handle URL loading within the app.
7. Implement an event listener for the "GET" Button to load the specified URL in the WebView.

Features used:

1. WebView for rendering HTML content.
2. EditText for user input of a URL.
3. Button to trigger the loading of the specified URL.
4. Statically defined HTML content for display.
5. WebViewClient to handle URL loading within the app.

Source code:

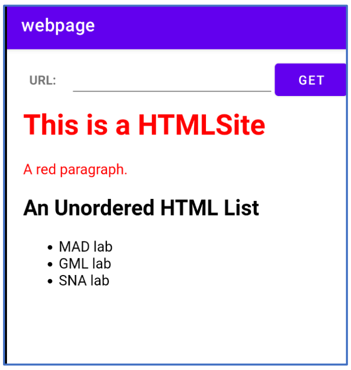
- MainActivity.java

```java
package com.example.webpage;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    private WebView webView;
    private EditText url;
    private Button getButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        webView = findViewById(R.id.webView);
        url = findViewById(R.id.url);
        getButton = findViewById(R.id.load);
        webView.getSettings().setJavaScriptEnabled(true);
        String staticHtml = "<html>\n" +
                "<body>\n" +
                "\n" +
                "<h1 style=\"color:red; font-family:sans-serif\">This is a HTML" +
                "Site</h1>\n" +
                "\n" +
                "<p style=\"color:red;\">A red paragraph.</p>\n" +
                "\n" +
                "<h2>An Unordered HTML List</h2>\n" +
                "\n" +
                "<ul>\n" +
                "  <li>MAD lab</li>\n" +
                "  <li>GML lab</li>\n" +
                "  <li>SNA lab</li>\n" +
                "</ul>\n" +
                "\n" +
                "</body>\n" +
                "</html>";
        webView.loadData(staticHtml, "text/html", "UTF-8");
        webView.setWebViewClient(new WebViewClient() {
            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url) {
                // view.loadUrl(url);
                System.out.println("hello");
                return false;
            }
        });
        getButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                webView.loadUrl("https://" + url.getText().toString());
            }
        });
    }

}
```

- activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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" android:padding="10dp" tools:context=".MainActivity">
    <LinearLayout android:layout_width="match_parent" android:layout_height="match_parent"
android:orientation="vertical">
        <LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content" android:orientation="horizontal">
            <TextView android:id="@+id/textView" android:layout_width="60dp"
android:layout_height="wrap_content" android:paddingBottom="10dp"
android:paddingLeft="15dp" android:textStyle="bold" android:text="URL:" />
            <EditText android:id="@+id/url" android:layout_width="237dp"
android:layout_height="wrap_content" android:layout_weight="1" android:ems="10"
android:inputType="text" />
            <Button android:id="@+id/load" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_weight="1" android:text="GET" />
        </LinearLayout>
        <WebView android:id="@+id/webView" android:layout_width="match_parent"
android:layout_height="match_parent">
        </WebView>
    </LinearLayout>
</LinearLayout>
```

Output:

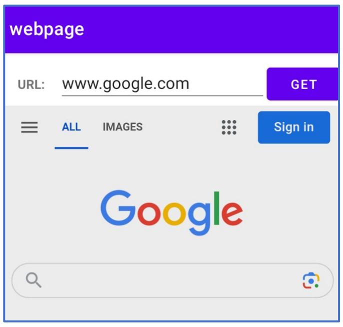


This is a HTMLSite

A red paragraph.

An Unordered HTML List

- MAD lab
- GML lab
- SNA lab



www.google.com

Result:

The mobile application was completed successfully

Best Practices:

1. Enable JavaScript in WebView settings when needed for improved web page compatibility.
2. Use constants or string resources for strings used in code to facilitate localization.
3. Implement a WebViewClient to handle URL loading within the app and prevent opening external browsers.

Learning Outcomes:

1. Understanding and implementing WebView in Android for displaying HTML content.
2. Enabling JavaScript in WebView settings for enhanced web page functionality.
3. Handling user input through EditText and triggering actions with buttons.
4. Implementing WebViewClient to control URL loading behavior.