

- ❖ **Course Module : MIT3206 - Mobile Computing**
- ❖ **Course Lecturer : Senior Lecturer Gihan P. Seneviratne Sir**

❖ Assignment 8 : Web View and Browsing Internal Storage

❖ **Assignment 8 Google Drive Uploaded Link :**

https://drive.google.com/drive/folders/1nstDB7WXqWA1odnx0db80egUM1KQ2I_x?usp=sharing

❖ **GitHub Private Repository Link :**

<https://github.com/NavinduMadusanka/Assignment8-MIT3206-22550119.git>

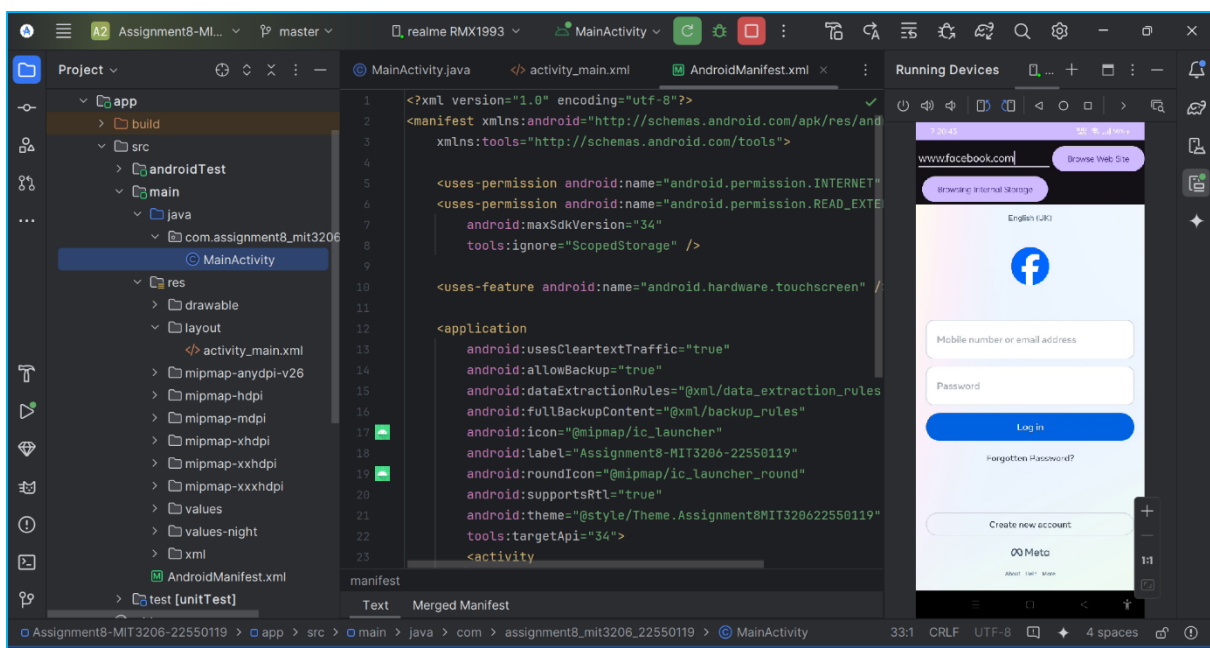
❖ **Student Name : Kumarage Navindu Madusanka Dias (K.N.M. Dias)**

❖ Student Index No : 22550119

❖ Student Registration No : 2022/MIT/011

❖ Email Address : navindu09@gmail.com

❖ **Contact No : 0702678624**



Assignment8-MIT3206-22550119

Assignment 8 : Web View and Browsing Internal Storage

Below is a summary of what I have learned and focused on in this assignment.

1. Android Components

- **API level**

Android Version	API Level	Version Name
Android 7.0	24	Nougat

- **Class**

WebView Class (An extension of Android's View Class)

- **Common Control Methods are using in WebView**

No	Method	Description
1	canGoBack()	WebView has a backward.
2	canGoForward()	WebView has a forward.
3	clearHistory()	Clear the WebView forward and backward history.
4	destroy()	Destroy the internal state of WebView.
5	getProgress()	Gets the progress of the current page.
6	getTitle()	Return the title of the current page.
7	getUrl()	Return the URL of the current page.

- **Permissions**
 - uses-permission – Access INTERNET
 - uses-permission – READ EXTERNAL STORAGE
- **Newly learned key points in this assignment**
 - Clicking on a link in the WebView site will not load that page into the WebView application.
 - The default browser has been loaded.
To achieve this, I have to extend and override WebViewClient class.
 - How to disable default web browser.
- **tools:targetApi="34" in AndroidManifest.xml**
- **android:usesCleartextTraffic="true" in AndroidManifest.xml**
- **uses-feature android:name="android.hardware.touchscreen"**

Component	Innovative Android Version	Minimum Supported Version	Impact on Android	Usability
Web View	Android 1.0	Android 1.0 (API Level 1)	WebView enables embedding web content within apps, bridging the gap between native and web experiences.	Essential for apps that need to display web content without leaving the app environment.

2. Functionality of the mobile application

- Just specify a URL on the URL field and press the “Browse Web Site” button that appears, to launch the website.
- Browse Local files - Touch on the “Browsing Internal Storage” button, to launch the Local files.

3. Running the Application on my android mobile device

I was running the android app for testing in my android mobile device.

My android mobile device is Realme X2 RMX1993.

Below is a Screenshots of my android mobile device (Realme X2 RMX1993) while the app was running.



Photo 1 : Assignment8-MIT3206-22550119 Web View in Realme X2 RMX1993

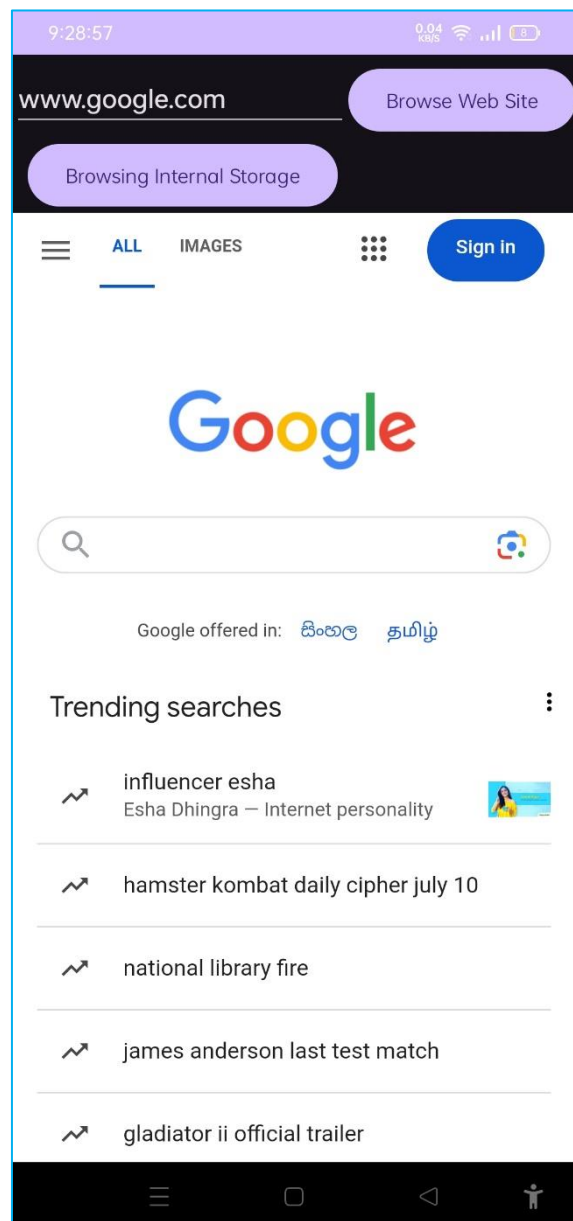


Photo 2 : Assignment8-MIT3206-22550119 Web View in Realme X2 RMX1993

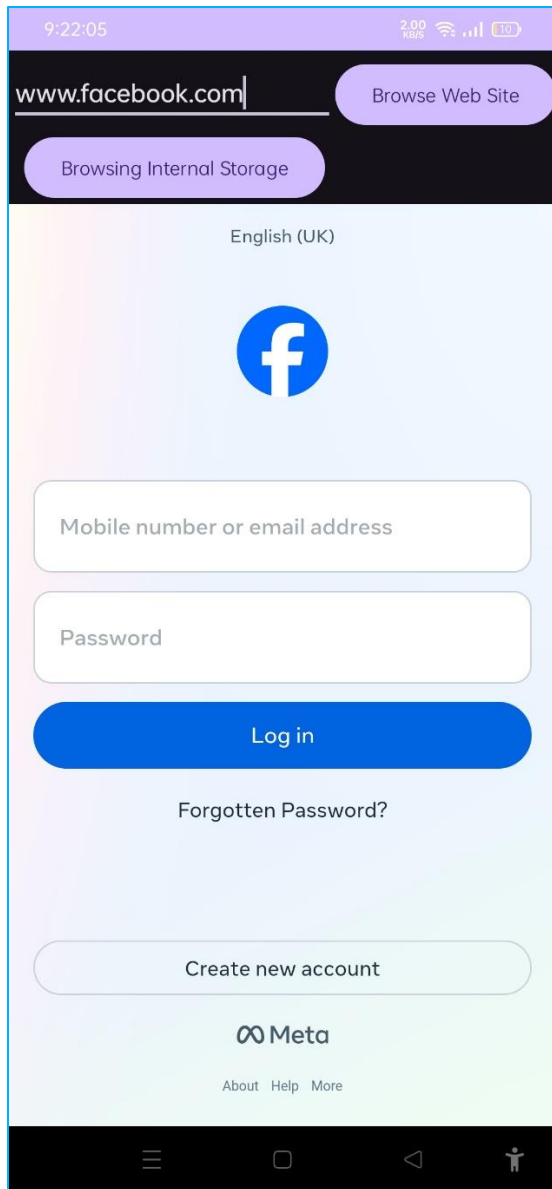


Photo 3 : Assignment8-MIT3206-22550119 Web View in Realme X2 RMX1993

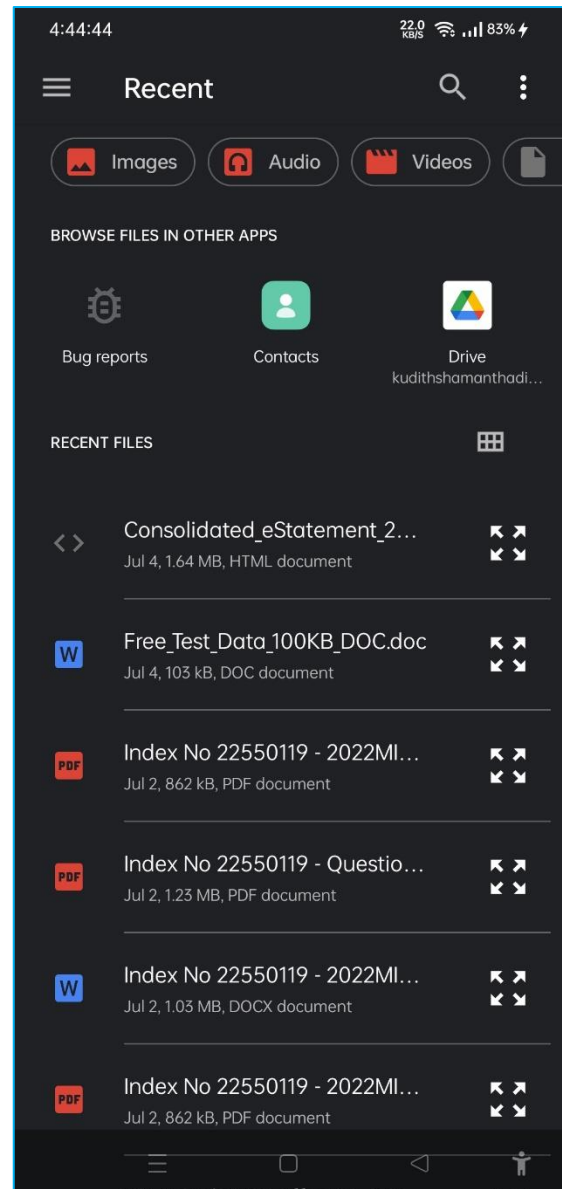


Photo 4 : Assignment8-MIT3206-22550119 Browsing Internal Storage in Realme X2 RMX1993

4. Main Coding files

- **MainActivity.java**

```
package com.assignment8_mit3206_22550119;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebChromeClient;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText urlEditText;
    private Button browseWebButton;
    private Button browseLocalFilesButton;
    private WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        urlEditText = findViewById(R.id.url_edit_text);
        browseWebButton = findViewById(R.id.browse_web_button);
        browseLocalFilesButton = findViewById(R.id.browse_local_files_button);
        webView = findViewById(R.id.web_view);

        WebSettings webSettings = webView.getSettings();
        webSettings.setJavaScriptEnabled(true);

        webView.setWebViewClient(new WebViewClient() {
            @Override
            public boolean shouldOverrideUrlLoading(WebView view, String url) {
                view.loadUrl(url);
                return true;
            }
        });
    }
}
```

```
webView.setWebChromeClient(new WebChromeClient());

browseWebButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String url = urlEditText.getText().toString();
        if (!url.startsWith("http://") && !url.startsWith("https://")) {
            url = "http://" + url;
        }
        webView.loadUrl(url);
    }
});

browseLocalFilesButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
        intent.setType("*/*");
        startActivityForResult(intent, 1);
    }
});

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 1 && resultCode == RESULT_OK) {
        Uri uri = data.getData();
        webView.loadUrl(uri.toString());
    }
}
}
```

- **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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"
        android:layout_marginTop="5dp">
```

```
<EditText
    android:id="@+id/url_edit_text"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:textStyle="bold"
    android:hint="Enter URL"
    android:inputType="textUri" />

<Button
    android:id="@+id/browse_web_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:text="Browse Web Site" />

</LinearLayout>

<Button
    android:id="@+id/browse_local_files_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textStyle="bold"
    android:layout_marginLeft="10dp"
    android:text="Browsing Internal Storage" />

<WebView
    android:id="@+id/web_view"
    android:layout_width="match_parent"
    android:textStyle="bold"
    android:layout_height="0dp"
    android:layout_weight="1" />

</LinearLayout>
```


- **AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
        android:maxSdkVersion="34"
        tools:ignore="ScopedStorage" />

    <uses-feature android:name="android.hardware.touchscreen" />

    <application
        android:usesCleartextTraffic="true"
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Assignment8-MIT3206-22550119"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Assignment8MIT320622550119"
        tools:targetApi="34">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```