apk-builder/   .github/workflows/build-apk.yml   app/   build.gradle
src/main/   AndroidManifest.xml
java/com/webview/app/MainActivity.java   res/   layout/activity_main.xml
    values/strings.xml   values/colors.xml   values/styles.xml
build.gradle   settings.gradle   gradle/wrapper/gradle-wrapper.properties
name: Build APK

on: workflow_dispatch: inputs: app_id: description: 'App ID from database' required:
true app_name: description: 'App display name' required: true website_url:
description: 'Website URL to wrap' required: true logo_url: description: 'Logo URL for
app icon' required: false webhook_url: description: 'Webhook URL for build
completion' required: true

jobs: build: runs-on: ubuntu-latest

```
  steps:
    - uses: actions/checkout@v4

    - name: Set up JDK 17
      uses: actions/setup-java@v4
      with:
        java-version: '17'
        distribution: 'temurin'

    - name: Replace placeholders
      run: |
        # Replace website URL
        find . -type f -name "*.java" -exec sed -i "s|WEBSITE_URL_PLACEHOLDER|${{ gith
        find . -type f -name "*.xml" -exec sed -i "s|APP_NAME_PLACEHOLDER|${{ github
        find . -type f -name "*.xml" -exec sed -i "s|WEBSITE_URL_PLACEHOLDER|${{ gith

        # Update strings.xml app name
        sed -i "s|APP_NAME_PLACEHOLDER|${{ github.event.inputs.app_name }}|g" ap

    - name: Download custom icon
      if: ${{ github.event.inputs.logo_url != '' }}
      run: |
        curl -L -o app/src/main/res/mipmap-hdpi/ic_launcher.png "${{ github.event.inpu
        cp app/src/main/res/mipmap-hdpi/ic_launcher.png app/src/main/res/mipmap-r
```

```
        cp app/src/main/res/mipmap-hdpi/ic_launcher.png app/src/main/res/mipmap->
        cp app/src/main/res/mipmap-hdpi/ic_launcher.png app/src/main/res/mipmap->

  - name: Grant execute permission for gradlew
    run: chmod +x gradlew

  - name: Build APK
    run: ./gradlew assembleRelease --no-daemon

  - name: Create Release
    id: create_release
    uses: actions/create-release@v1
    env:
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
    with:
      tag_name: build-${{ github.run_id }}
      release_name: "APK - ${{ github.event.inputs.app_name }}"
      draft: false
      prerelease: false

  - name: Upload APK to Release
    id: upload_apk
    uses: actions/upload-release-asset@v1
    env:
      GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
    with:
      upload_url: ${{ steps.create_release.outputs.upload_url }}
      asset_path: app/build/outputs/apk/release/app-release-unsigned.apk
      asset_name: ${{ github.event.inputs.app_name }}.apk
      asset_content_type: application/vnd.android.package-archive

  - name: Notify webhook
    if: always()
    run: |
      STATUS="completed"
      APK_URL="${{ steps.upload_apk.outputs.browser_download_url }}"
      if [ "${{ job.status }}" != "success" ]; then
        STATUS="failed"
        APK_URL=""
```

```
    fi
    curl -X POST "${{ github.event.inputs.webhook_url }}" \
      -H "Content-Type: application/json" \
      -d "{\"app_id\": \"${{ github.event.inputs.app_id }}\", \"status\": \"${STATUS}\", \
```

buildscript { repositories { google() mavenCentral() } dependencies { classpath 'com.android.tools.build:gradle:8.2.0' } }

allprojects { repositories { google() mavenCentral() } }

task clean(type: Delete) { delete rootProject.buildDir } pluginManagement { repositories { google() mavenCentral() gradlePluginPortal() } }

rootProject.name = "WebViewApp" include ':app' plugins { id 'com.android.application' }

android { namespace 'com.webview.app' compileSdk 34

```
  defaultConfig {
      applicationId "com.webview.app"
      minSdk 24
      targetSdk 34
      versionCode 1
      versionName "1.0"
  }

  buildTypes {
      release {
          minifyEnabled false
          proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'progua
      }
  }

  compileOptions {
      sourceCompatibility JavaVersion.VERSION_17
      targetCompatibility JavaVersion.VERSION_17
  }
```

}

dependencies { implementation 'androidx.appcompat:appcompat:1.6.1'
implementation 'androidx.webkit:webkit:1.8.0' implementation
'androidx.swiperefreshlayout:swiperefreshlayout:1.1.0' }

```xml
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" /

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="APP_NAME_PLACEHOLDER"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    android:usesCleartextTraffic="true">

    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:configChanges="orientation|screenSize|keyboardHidden">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

package com.webview.app;

import android.annotation.SuppressLint; import android.content.Intent; import
android.graphics.Bitmap; import android.net.Uri; import android.os.Bundle; import
android.view.KeyEvent; import android.view.View; import android.webkit.*; import
android.widget.ProgressBar; import androidx.appcompat.app.AppCompatActivity;
import androidx.swiperefreshlayout.widget.SwipeRefreshLayout;

public class MainActivity extends AppCompatActivity {

```java
private WebView webView;
private ProgressBar progressBar;
private SwipeRefreshLayout swipeRefresh;
```

```java
private static final String WEBSITE_URL = "WEBSITE_URL_PLACEHOLDER";

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

    webView = findViewById(R.id.webView);
    progressBar = findViewById(R.id.progressBar);
    swipeRefresh = findViewById(R.id.swipeRefresh);

    // WebView Settings
    WebSettings settings = webView.getSettings();
    settings.setJavaScriptEnabled(true);
    settings.setDomStorageEnabled(true);
    settings.setLoadWithOverviewMode(true);
    settings.setUseWideViewPort(true);
    settings.setAllowFileAccess(true);
    settings.setAllowContentAccess(true);
    settings.setCacheMode(WebSettings.LOAD_DEFAULT);
    settings.setMixedContentMode(WebSettings.MIXED_CONTENT_ALWAYS_ALLOW)

    webView.setWebViewClient(new WebViewClient() {
        @Override
        public void onPageStarted(WebView view, String url, Bitmap favicon) {
            progressBar.setVisibility(View.VISIBLE);
        }

        @Override
        public void onPageFinished(WebView view, String url) {
            progressBar.setVisibility(View.GONE);
            swipeRefresh.setRefreshing(false);
        }

        @Override
        public boolean shouldOverrideUrlLoading(WebView view, WebResourceReque
            String url = request.getUrl().toString();
            if (url.startsWith("http://") || url.startsWith("https://")) {
```

```java
            if (url.contains(Uri.parse(WEBSITE_URL).getHost())) {
                return false; // Load in WebView
            } else {
                // Open external links in browser
                Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
                startActivity(intent);
                return true;
            }
        }
        return false;
    }
});

webView.setWebChromeClient(new WebChromeClient() {
    @Override
    public void onProgressChanged(WebView view, int newProgress) {
        progressBar.setProgress(newProgress);
    }
});

// Swipe to refresh
swipeRefresh.setOnRefreshListener(() -> webView.reload());

webView.loadUrl(WEBSITE_URL);
}

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (keyCode == KeyEvent.KEYCODE_BACK && webView.canGoBack()) {
        webView.goBack();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

} <androidx.swiperefreshlayout.widget.SwipeRefreshLayout xmlns:android="http://schemas.android.com/apk/res/android" android:id="@+id/swipeRefresh" android:layout_width="match_parent" android:layout_height="match_parent">

```xml
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ProgressBar
        android:id="@+id/progressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="4dp"
        android:layout_alignParentTop="true"
        android:indeterminate="false"
        android:max="100"
        android:visibility="gone" />

    <WebView
        android:id="@+id/webView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/progressBar" />

</RelativeLayout>
```

```
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
APP_NAME_PLACEHOLDER #1976D2 #1565C0 #FF4081 #FFFFFF
distributionBase=GRADLE_USER_HOME distributionPath=wrapper/dists
distributionUrl=https://services.gradle.org/distributions/gradle-8.2-bin.zip
zipStoreBase=GRADLE_USER_HOME zipStorePath=wrapper/dists
```