EXPERIMENT No:7

AIM: To test and deploy production ready Flutter App on Android platform

THEORY:

Build and release an Android app

During a typical development cycle, you test an app using flutter run at the command line, or by using the **Run** and **Debug** options in your IDE. By default, Flutter builds a *debug* version of your app.
To publish an app to the Google Play Store, you need to prepare a *release* version of your app. Before publishing, you might want to put some finishing touches on your app. This may covers the following topics:

- Adding a launcher icon
- Enabling Material Components
- Signing the app
- Shrinking your code with R8
- Enabling multidex support
- Reviewing the app manifest
- Reviewing the build configuration
- Building the app for release
- Publishing to the Google Play Store
- Updating the app's version number
- Android release FAQ

Note: Throughout this manual, [project] refers to the directory that your application is in. While following these instructions, substitute [project] with your app's directory.

Adding a launcher icon

When a new Flutter app is created, it has a default launcher icon. To customize this icon, you might want to check out the flutter_launcher_icons package.

Alternatively, you can do it manually using the following steps:

1. Review the Material Design product icons guidelines for icon design.
2. In the [project]/android/app/src/main/res/ directory, place your icon files in folders named using configuration qualifiers. The default mipmap- folders demonstrate the correct naming convention.
3. In AndroidManifest.xml, update the application tag's android:icon attribute to reference icons from the previous step (for example, <application android:icon="@mipmap/ic_launcher" ...).
4. To verify that the icon has been replaced, run your app and inspect the app icon in the Launcher.

Enabling Material Components

If your app uses Platform Views, you may want to enable Material Components by following the steps described in the Getting Started guide for Android.

For example:

1. Add the dependency on Android's Material in <my-app>/android/app/build.gradle:

```
dependencies {
   implementation 'com.google.android.material:material:<version>'
 }
```

To find out the latest version, visit Google Maven.

Signing the app

To publish on the Play Store, you need to give your app a digital signature. Use the following instructions to sign your app.

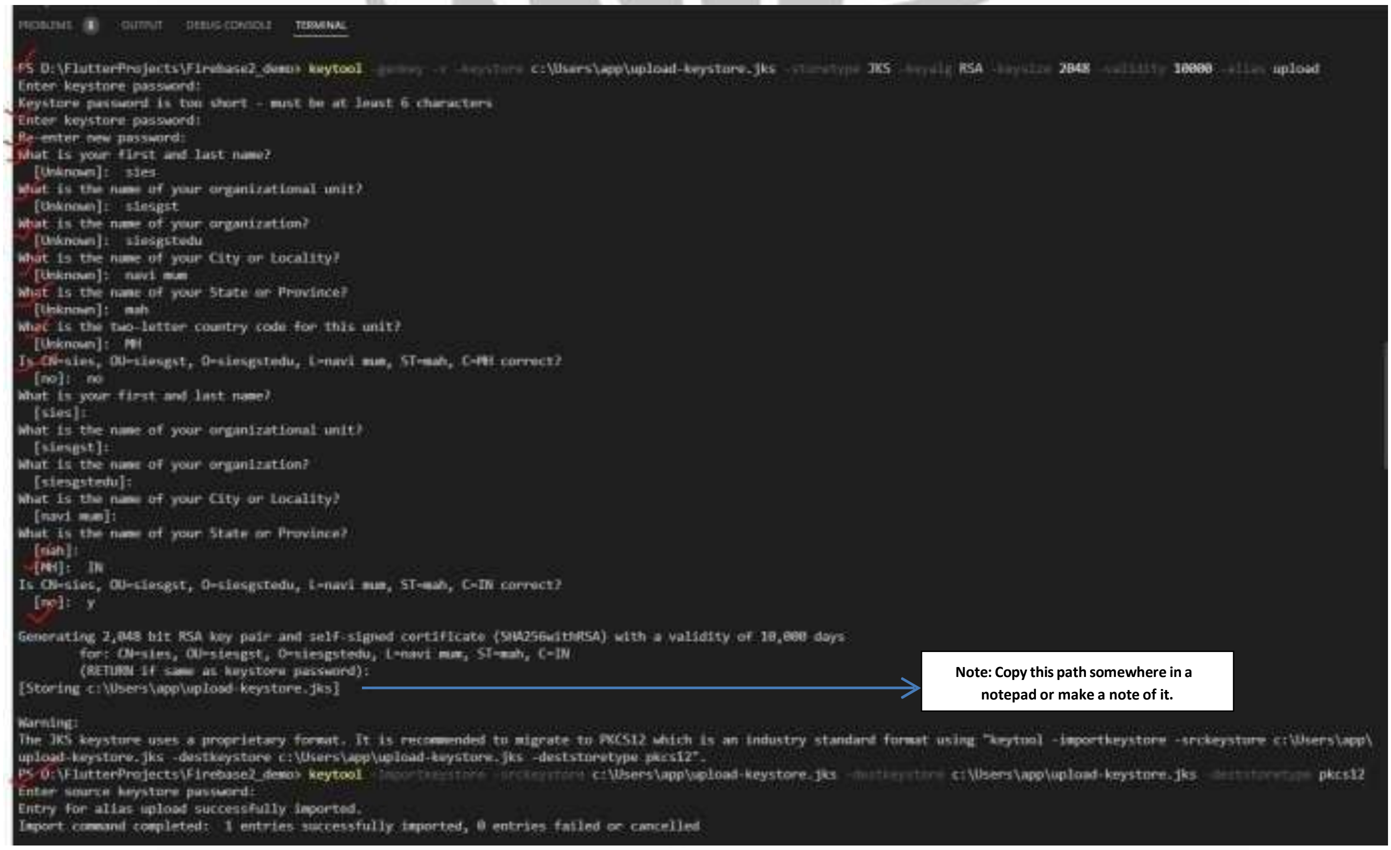
On Android, there are two signing keys: **deployment and upload**. The end-users download the .apk signed with the ‘deployment key’. An ‘upload key’ is used to authenticate the .aab / .apk uploaded by developers onto the Play Store and is re-signed with the deployment key once in the Play Store.

- It’s highly recommended to use the automatic cloud managed signing for the deployment key. For more information, see the official Play Store documentation.

On Windows, use the following command:

```
keytool -genkey -v -keystore c:\Users\USER_NAME\upload-keystore.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias upload
```

This command stores the upload-keystore.jks file in your home directory. If you want to store it elsewhere, change the argument you pass to the -keystore parameter. **However, keep the keystore file private; don’t check it into**



```
PS D:\FlutterProjects\Firebase2_demo> keytool -genkey -v -keystore c:\Users\app\upload-keystore.jks -storetype JKS -keyalg RSA -keysize 2048 -validity 10000 -alias upload
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  sies
What is the name of your organizational unit?
  [Unknown]:  siesgst
What is the name of your organization?
  [Unknown]:  siesgstedu
What is the name of your City or Locality?
  [Unknown]:  navi mum
What is the name of your State or Province?
  [Unknown]:  mah
What is the two-letter country code for this unit?
  [Unknown]:  MH
Is CN=sies, OU=siesgst, O=siesgstedu, L=navi mum, ST=mah, C=MH correct?
  [no]:  no
What is your first and last name?
  [sies]:
What is the name of your organizational unit?
  [siesgst]:
What is the name of your organization?
  [siesgstedu]:
What is the name of your City or Locality?
  [navi mum]:
What is the name of your State or Province?
  [mah]:
  [MH]:  IN
Is CN=sies, OU=siesgst, O=siesgstedu, L=navi mum, ST=mah, C=IN correct?
  [no]:  y

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
        for: CN=sies, OU=siesgst, O=siesgstedu, L=navi mum, ST=mah, C=IN
        (RETURN if same as keystore password):
[Storing c:\Users\app\upload-keystore.jks]

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkeystore -srckeystore c:\Users\app\
upload-keystore.jks -destkeystore c:\Users\app\upload-keystore.jks -deststoretype pkcs12".
PS D:\FlutterProjects\Firebase2_demo> keytool -importkeystore -srckeystore c:\Users\app\upload-keystore.jks -destkeystore c:\Users\app\upload-keystore.jks -deststoretype pkcs12
Enter source keystore password:
Entry for alias upload successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled
```

Note: Copy this path somewhere in a notepad or make a note of it.

Create an upload keystore

If you have an existing keystore, skip to the next step. If not, create one by either:

- Following the Android Studio key generation steps
- Running the following at the command line:

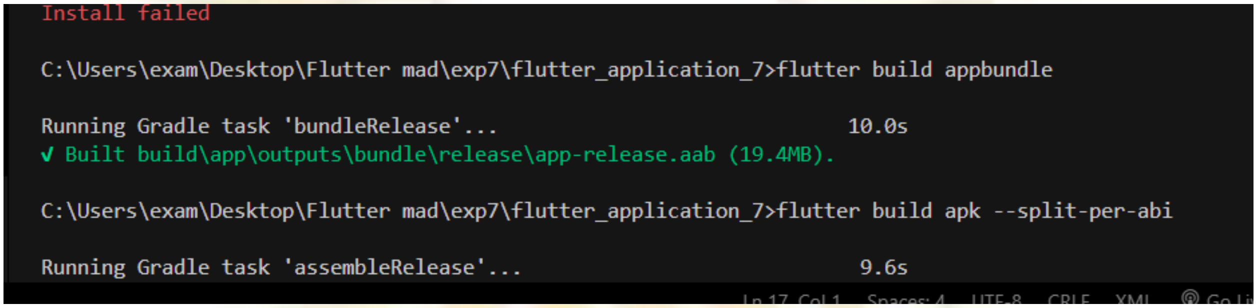
On Mac/Linux, use the following command:

```
keytool -genkey -v -keystore ~/upload-keystore.jks -keyalg RSA -keysize 2048 -validity 10000 -alias
upload
```

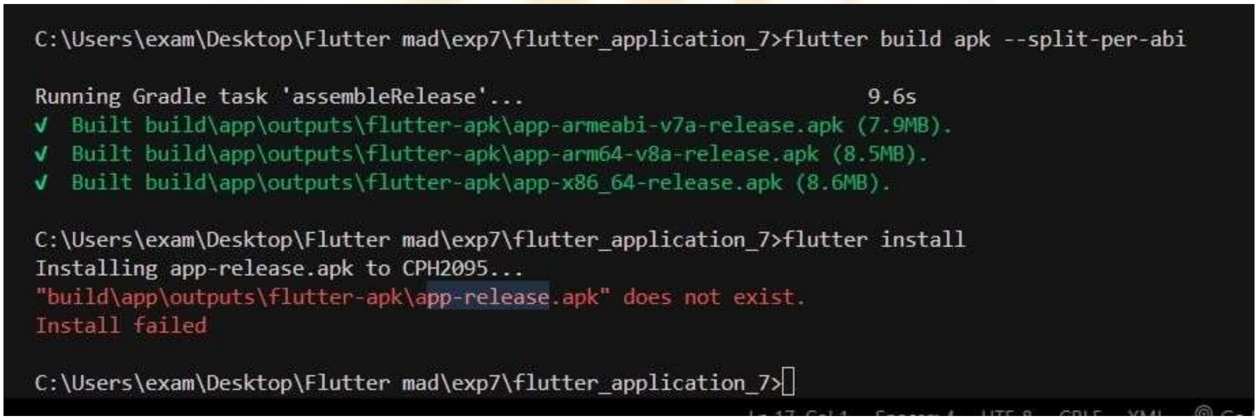
Note:

- The keytool command might not be in your path—it's part of Java, which is installed as part of Android Studio. For the concrete path, run flutter doctor -v and locate the path printed after 'Java binary at:' then use that fully qualified path replacing java (at the end) with keytool. If your path includes space-separated names, such as Program Files, use platform-appropriate notation for the names. For example, on Mac/Linux use Program\ Files, and on Windows use "Program Files".
- The -storetype JKS tag is only required for Java 9 or newer. As of the Java 9 release, the keystore type defaults to PKS12.

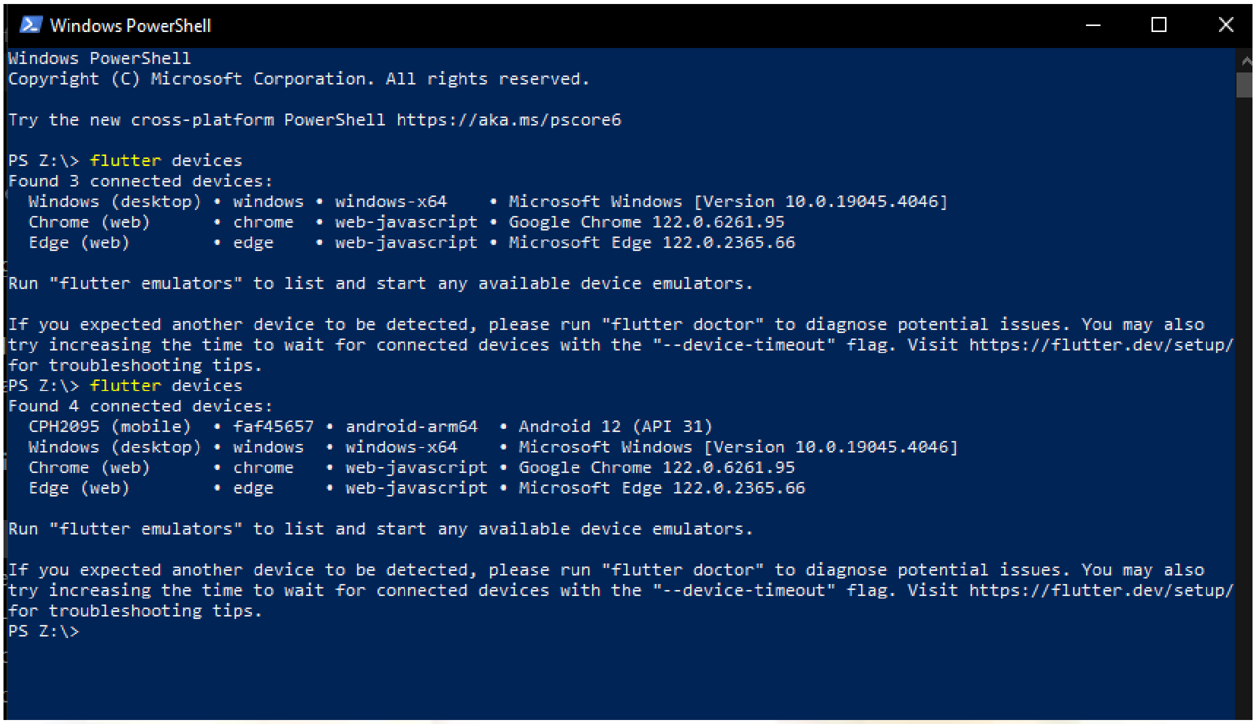
OUTPUT:



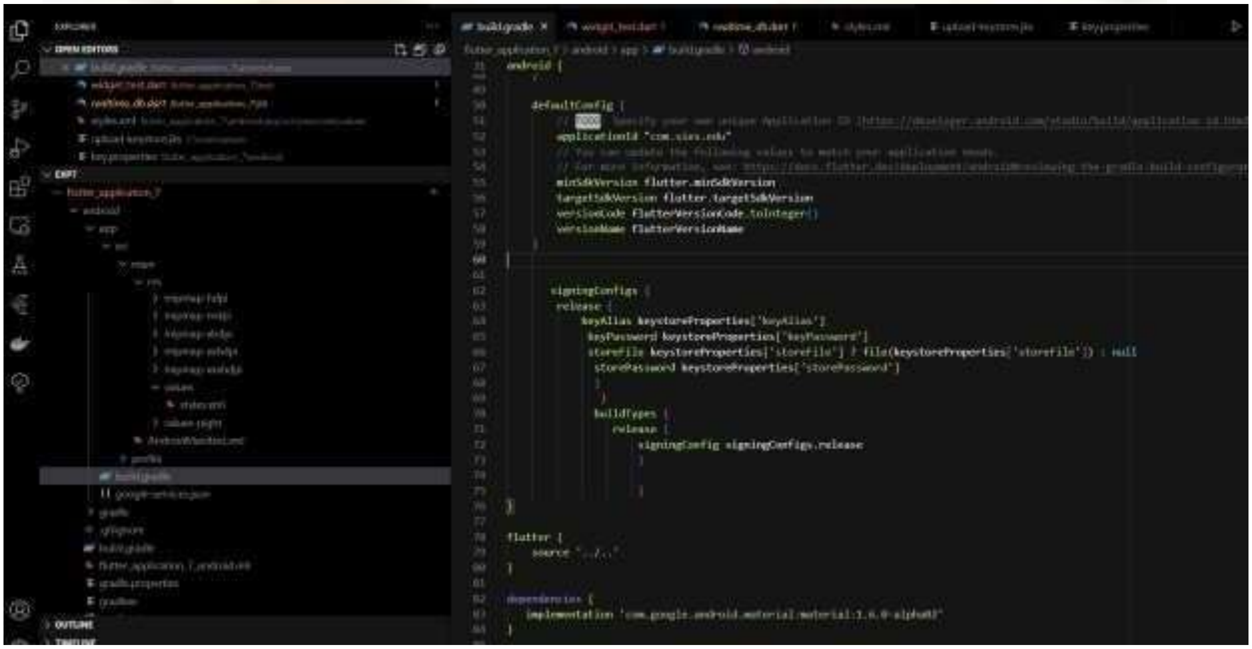
```
Install failed

C:\Users\exam\Desktop\Flutter mad\exp7\flutter_application_7>flutter build appbundle

Running Gradle task 'bundleRelease'...                             10.0s
√ Built build\app\outputs\bundle\release\app-release.aab (19.4MB).

C:\Users\exam\Desktop\Flutter mad\exp7\flutter_application_7>flutter build apk --split-per-abi

Running Gradle task 'assembleRelease'...                            9.6s
```



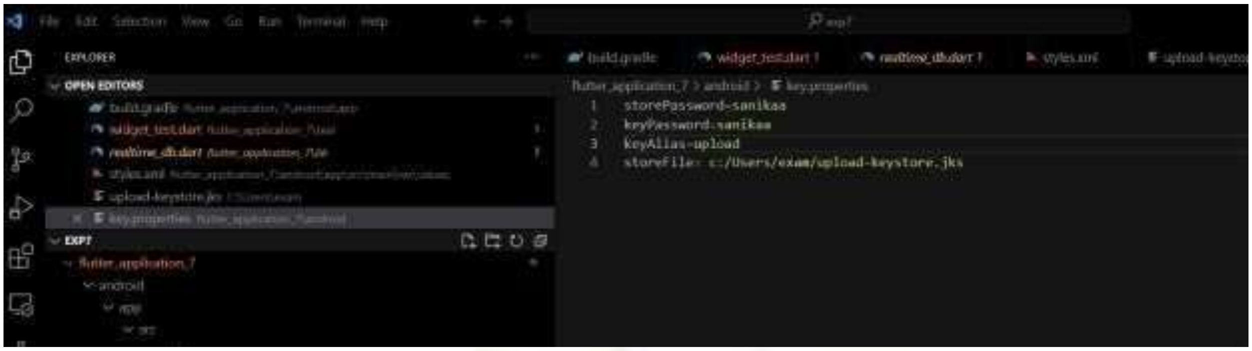
```
C:\Users\exam\Desktop\Flutter mad\exp7\flutter_application_7>flutter build apk --split-per-abi

Running Gradle task 'assembleRelease'...                            9.6s
√  Built build\app\outputs\flutter-apk\app-armeabi-v7a-release.apk (7.9MB).
√  Built build\app\outputs\flutter-apk\app-arm64-v8a-release.apk (8.5MB).
√  Built build\app\outputs\flutter-apk\app-x86_64-release.apk (8.6MB).

C:\Users\exam\Desktop\Flutter mad\exp7\flutter_application_7>flutter install
Installing app-release.apk to CPH2095...
"build\app\outputs\flutter-apk\app-release.apk" does not exist.
Install failed

C:\Users\exam\Desktop\Flutter mad\exp7\flutter_application_7>
```



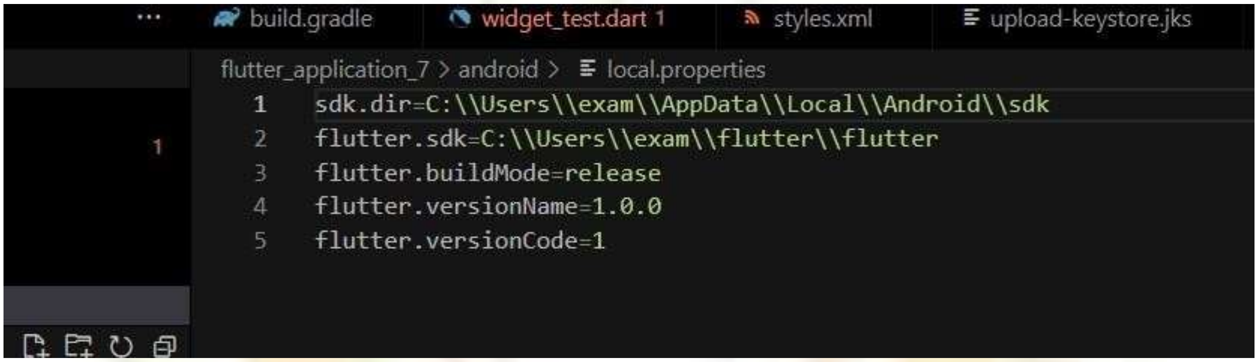
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS Z:\> flutter devices
Found 3 connected devices:
  Windows (desktop) • windows • windows-x64    • Microsoft Windows [Version 10.0.19045.4046]
  Chrome (web)      • chrome  • web-javascript • Google Chrome 122.0.6261.95
  Edge (web)        • edge    • web-javascript • Microsoft Edge 122.0.2365.66

Run "flutter emulators" to list and start any available device emulators.

If you expected another device to be detected, please run "flutter doctor" to diagnose potential issues. You may also
try increasing the time to wait for connected devices with the "--device-timeout" flag. Visit https://flutter.dev/setup/
for troubleshooting tips.
PS Z:\> flutter devices
Found 4 connected devices:
  CPH2095 (mobile)  • faf45657 • android-arm64  • Android 12 (API 31)
  Windows (desktop) • windows  • windows-x64    • Microsoft Windows [Version 10.0.19045.4046]
  Chrome (web)      • chrome   • web-javascript • Google Chrome 122.0.6261.95
  Edge (web)        • edge     • web-javascript • Microsoft Edge 122.0.2365.66

Run "flutter emulators" to list and start any available device emulators.

If you expected another device to be detected, please run "flutter doctor" to diagnose potential issues. You may also
try increasing the time to wait for connected devices with the "--device-timeout" flag. Visit https://flutter.dev/setup/
for troubleshooting tips.
PS Z:\>
```



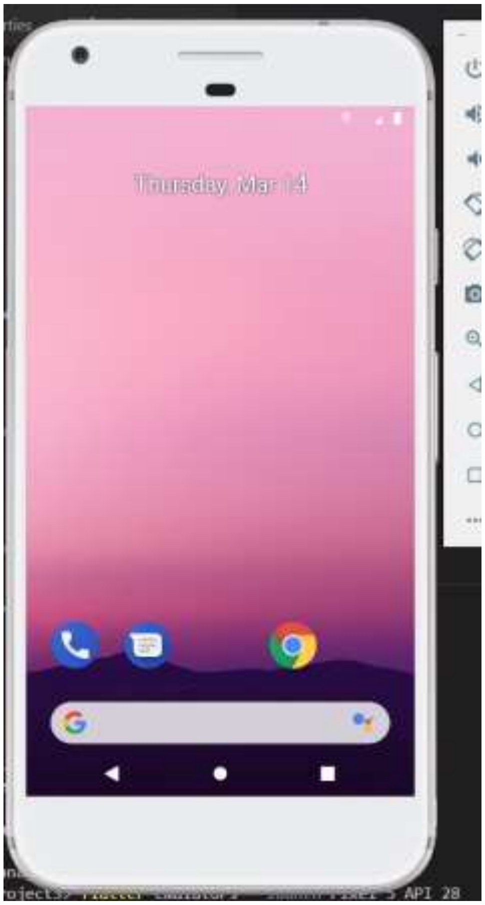
```
android {

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html
        applicationId "com.sies.edu"
        // You can update the following values to match your application needs.
        // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configurat
        minSdkVersion flutter.minSdkVersion
        targetSdkVersion flutter.targetSdkVersion
        versionCode flutterVersionCode.toInteger()
        versionName flutterVersionName
    }


      signingConfigs {
       release {
           keyAlias keystoreProperties['keyAlias']
            keyPassword keystoreProperties['keyPassword']
            storeFile keystoreProperties['storeFile'] ? file(keystoreProperties['storeFile']) : null
             storePassword keystoreProperties['storePassword']
             }
              }
             buildTypes {
                 release {
                     signingConfig signingConfigs.release
                     }

                     }
}

flutter {
    source '../..'
}

dependencies {
   implementation 'com.google.android.material:material:1.6.0-alpha02'
}
```



```
storePassword=sanikaa
keyPassword=sanikaa
keyAlias=upload
storeFile= c:/Users/exam/upload-keystore.jks
```



```
sdk.dir=C:\\Users\\exam\\AppData\\Local\\Android\\sdk
flutter.sdk=C:\\Users\\exam\\flutter\\flutter
flutter.buildMode=release
flutter.versionName=1.0.0
flutter.versionCode=1
```



Conclusion: We have successfully tested and deployed production ready Flutter App on Android platform.