How To Run Project

1. Flutter Installation Steps :

Windows

1. **Download Flutter SDK**:
    - Go to the official Flutter website: https://docs.flutter.dev/get-started/install/windows/mobile#install-the-flutter-sdk .
    - Download the latest Flutter SDK as a `.zip` file.
2. **Extract the Flutter SDK**:
    - Extract the zip file to a location such as `C:\src\flutter` (avoid paths with spaces or special characters).
3. **Update Path**:
    - Add `C:\src\flutter\bin` to the system `PATH`.
        - Right-click *This PC* > *Properties* > *Advanced system settings* > *Environment Variables*.
        - Under *User variables*, select `Path` and click *Edit*.
        - Add the Flutter `bin` directory path.

Linux

1. **Download Flutter SDK**:
    - Visit the official Flutter website: https://docs.flutter.dev/get-started/install/linux/android#download-then-install-flutter .
    - Download the latest Flutter SDK `.tar.xz` file.
2. **Extract Flutter SDK**:

Extract the file using the following command:

```
tar xf ~/Downloads/flutter_linux_v1.x.x-stable.tar.xz
```

Move the extracted folder to a suitable directory, e.g., `~/development/flutter`.

3. **Update Path**:
   `export PATH="$PATH:$HOME/development/flutter/bin"`
   - Add this line to your shell configuration file (`.bashrc`, `.zshrc`, etc.) for persistent use:
     `echo 'export PATH="$PATH:$HOME/development/flutter/bin"' >> ~/.bashrc`
   - Apply changes:
     `source ~/.bashrc`
4. **Install Dependencies**:

Install required system packages for Flutter:

`sudo apt-get install clang cmake ninja-build pkg-config libgtk-3-dev liblzma-dev`

3. Setting up Flutter in Visual Studio Code

1. **Install Visual Studio Code**:
   - Download and install VS Code from : https://code.visualstudio.com/ .
2. **Install Flutter & Dart Extensions**:
   - Open VS Code and go to the *Extensions* view by clicking on the Extensions icon or pressing Ctrl+Shift+X.
   - Search for **Flutter** and click **Install**. This will also install the **Dart** extension.
3. **Verify SDK Installation in VS Code**:
   - Press Ctrl+Shift+P to open the command palette.
   - Type Flutter: New Project to start a new Flutter project and ensure Flutter is correctly recognized by VS Code.

4. After installation, check the state of flutter:

    ```
    flutter doctor
    ```

    This command will check your environment and report any missing dependencies or issues. Follow the suggestions to resolve any issues (e.g., missing Android SDK, missing licenses).

Common Issues

Android licenses not accepted: Run the following command to resolve Android SDK license issues:

```
flutter doctor --android-licenses
```

Make Flutter recognize your phone as a device for development :

1. Enable Developer Options and USB Debugging on Your Phone

Android Devices:

1. **Enable Developer Options**:
    - Go to your phone's *Settings*.
    - Scroll down and tap on *About phone*.
    - Find the *Build number* and tap it **7 times** to enable Developer Options.
2. **Enable USB Debugging**:
    - Go back to *Settings*, and you'll now see *Developer options*.
    - Open *Developer options* and find *USB Debugging*. Enable it.

---

2. Connect Your Phone to Your Computer

- Use a USB cable to connect your phone to your computer.
- You should see a prompt on your phone asking if you want to allow USB debugging from the computer. Accept this prompt.

---

3. Set Up ADB (Android Debug Bridge)

For Windows:

1. **Install Android USB Drivers**:
    - Download the drivers for your specific phone manufacturer from https://developer.android.com/studio/run/oem-usb#Drivers .
    - Follow the installation instructions to install the drivers.
2. **Verify ADB**:
    - Open the Command Prompt (press Windows+R, type cmd, and hit Enter).

      Run the following command:

      ```
      adb devices
      ```

If ADB is properly installed and your phone is connected, it should show your device's serial number.

For Linux:

1. **Install ADB**:

Run the following command in the terminal:

```
sudo apt-get install adb
```

2. **Verify ADB**:

After installing ADB, verify your device by running:

```
adb devices
```

If successful, your phone should be listed as a connected device.

---

4. Restart VS Code and Check for Devices

1. Open **VS Code**.

In the terminal, run the following command to check for connected devices:

```
flutter devices
```

Your connected phone should be listed. If not, check the cable connection or make sure USB debugging is enabled on the phone.

2. If your phone is listed, you can now run Flutter apps directly on the device by using:

```
flutter run
```

. How To run Project :

1. Open the Project Directory

Navigate to the root folder of your Flutter project.

For example:

```
cd /path/to/your/flutter_project
```

2. Fetch Dependencies

Ensure that all necessary dependencies are installed:

```
flutter pub get
```

This will fetch the required packages listed in the pubspec.yaml file.

3. Run the App

Now, you can run your Flutter project:

Command for running on a connected device or emulator:

```
flutter run
```

This command will automatically detect a connected device or running emulator and launch the app.

Important Step :

I changed the tflite package code to make it work on the model I used. Since Flutter's package manager, flutter pub, downloads dependencies into the .pub-cache directory. This directory is located within your home directory. The exact path can vary depending on your operating system:

- **Linux/macOS**: ~/.pub-cache
- **Windows**: C:\Users\<YourUsername>\.pub-cache

Just replace the file in that directory named

On Linux:
/hosted/pub.dev/tflite_v2-1.0.0/android/src/main/java/sq/flutter/tflite/TflitePlugin.java

With the file with the same name “TflitePlugin.java” that is under the folder “extra”.

Other changes :

Inside the .pub-cache directory :

- **Linux/macOS**: ~/.pub-cache
- **Windows**: C:\Users\<YourUsername>\.pub-cache

➔ Replace the build.gradle file for **tflite_v2-1.0.0** package **:**

Go to : /hosted/pub.dev/tflite_v2-1.0.0/android/ and replace the file “build.gradle” with the file ‘build.gradle’ that is inside ‘extra/tflite’.

➔ Replace the build.gradle file for **flutter_tesseract_ocr-0.4.26** package **:**

Go to :/hosted/pub.dev/flutter_tesseract_ocr-0.4.26/android/ and replace the file “build.gradle” with the file ‘build.gradle’ that is inside ‘extra/flutter_tesseract_ocr’.

Create The APK:

Use the command :

```
flutter build apk
```

The apk will be found inside :

```
    build/app/outputs/flutter-apk/app-release.apk
```