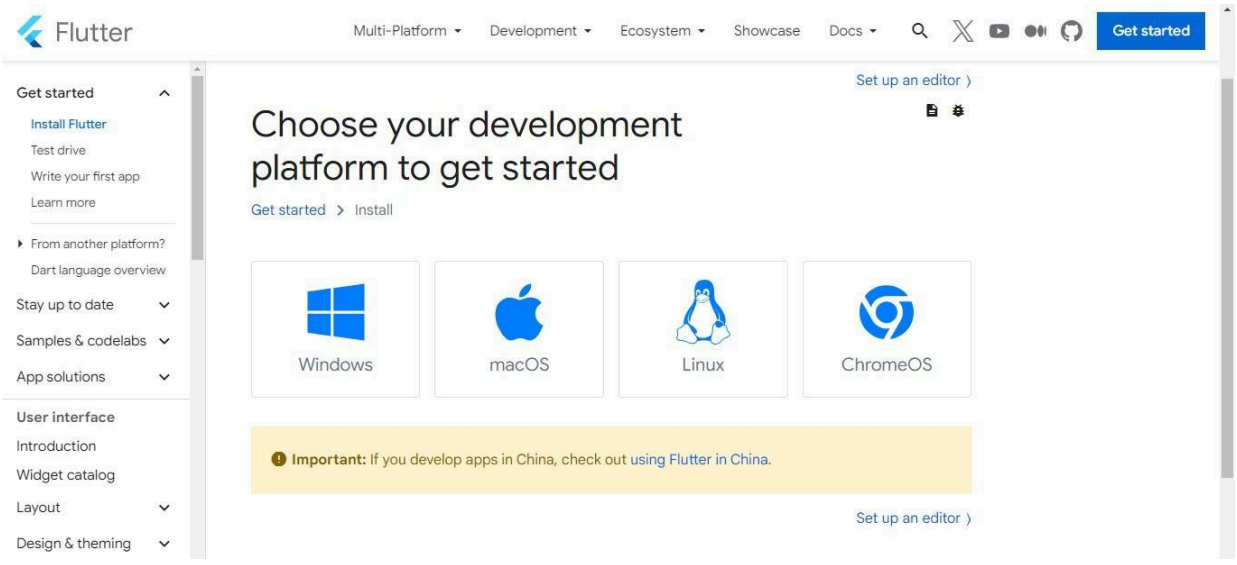
Experiment 1

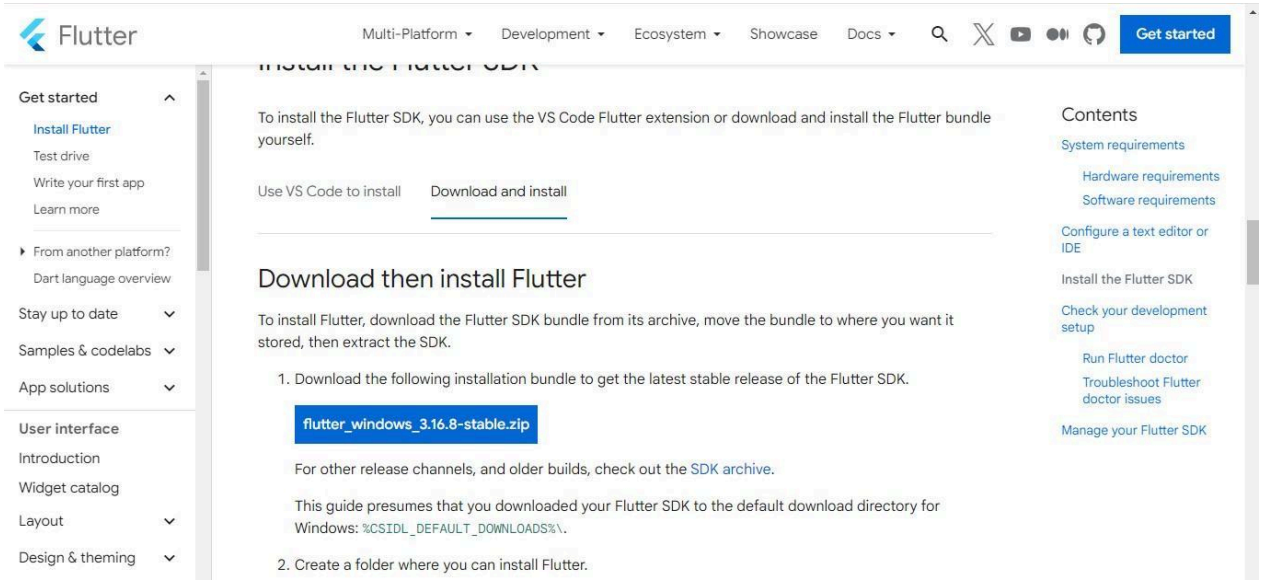
Aim: Installation and Configuration of Flutter Environment.

Theory:
Flutter, Google's toolkit for making apps on different devices, is changing how we create apps that work on iOS, Android, Desktop, and the Web. The key to Flutter is something called Widgets, which are like building blocks for making user interfaces that look good and work well. Flutter follows a consistent design style, no matter where the app runs. What's cool is that developers can mix and match these Widgets to create new things, making the development process easier and faster. Flutter is versatile, adapting smoothly to different platforms, so developers don't have to write separate code for each. The tree-like structure of widgets in Flutter makes it organized and easier to manage code, and the quick 'hot reload' feature lets developers see changes instantly, making the whole process faster and more efficient. In short, Flutter simplifies and speeds up app development, making it a smart choice for building apps that look great and work seamlessly on various devices.

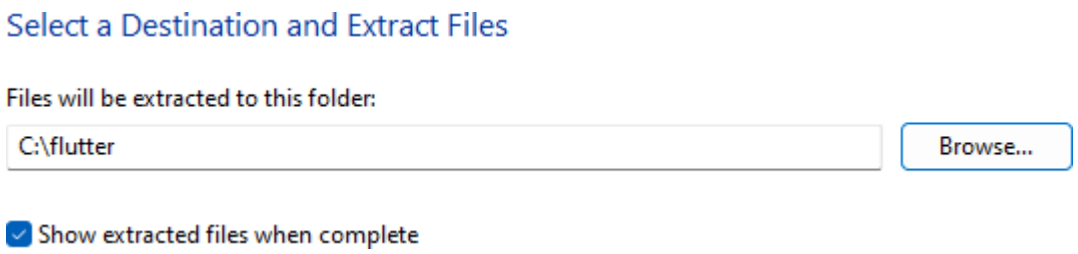
Install the Flutter SDK
Step 1: Download the installation bundle of the Flutter Software Development Kit for windows. To download Flutter SDK, Go to its official website https://docs.flutter.dev/get-started/install , you will get the following screen.



Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for SDK.

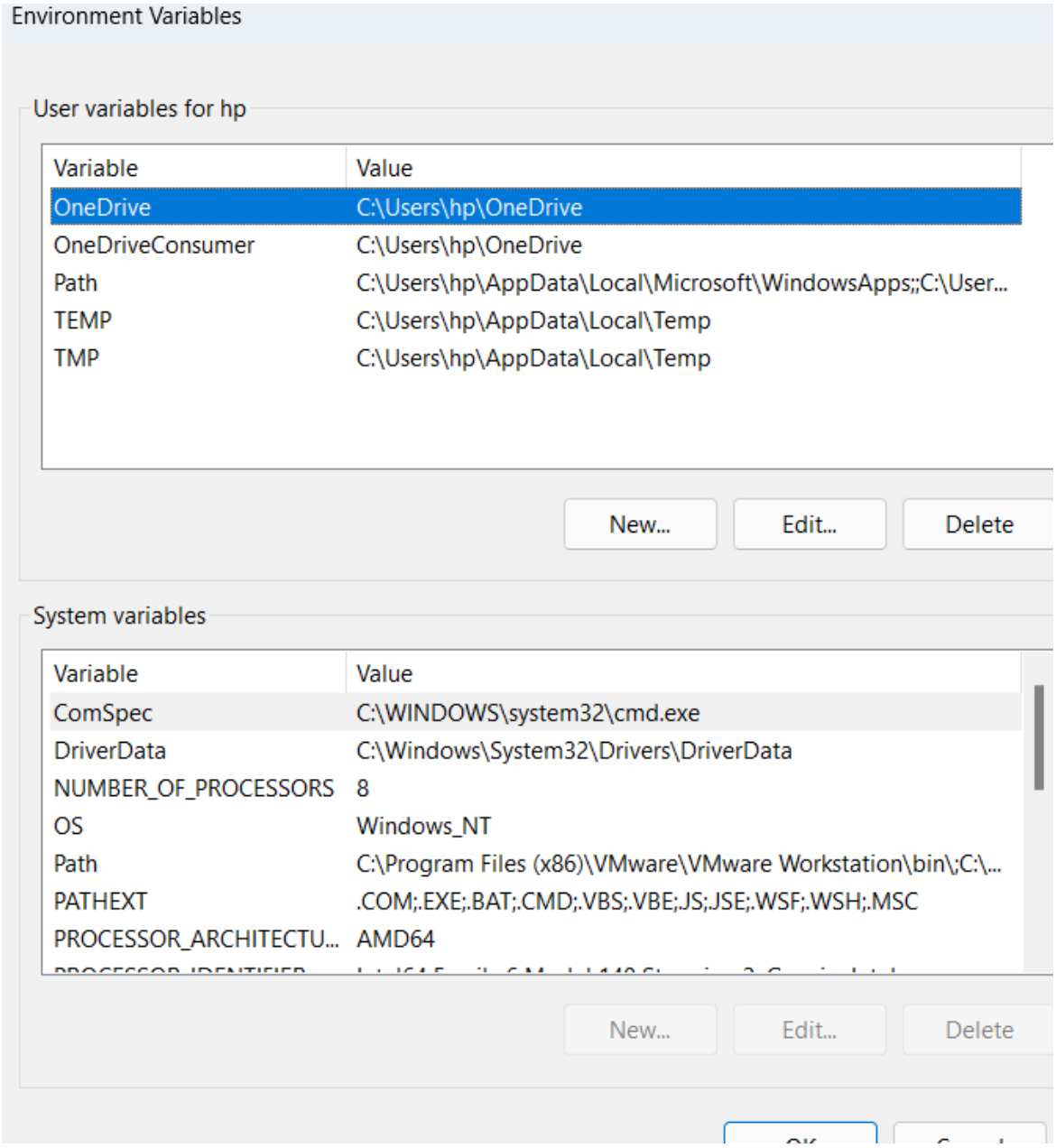


Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter.



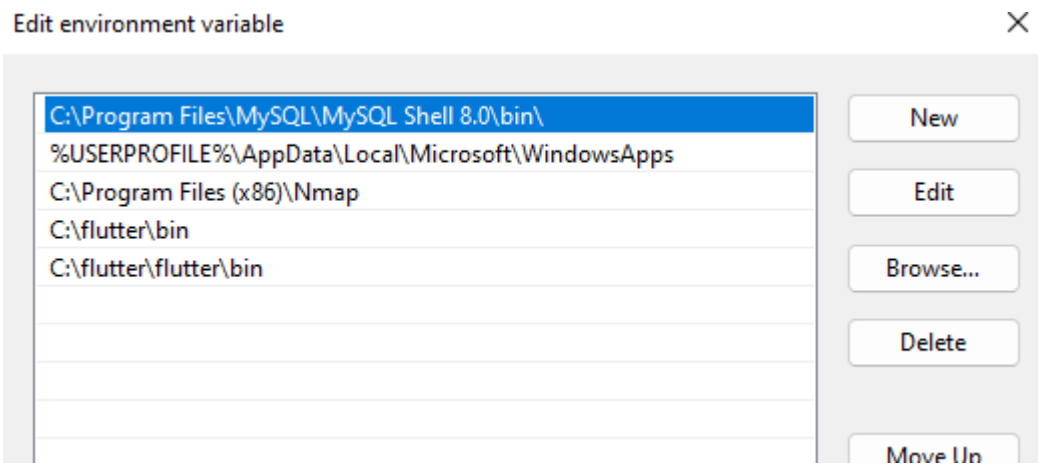
Step 4: To run the Flutter command in regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:

Step 4.1: Go to MyComputer properties ->; advanced tab -> environment variables. You will get the following screen.

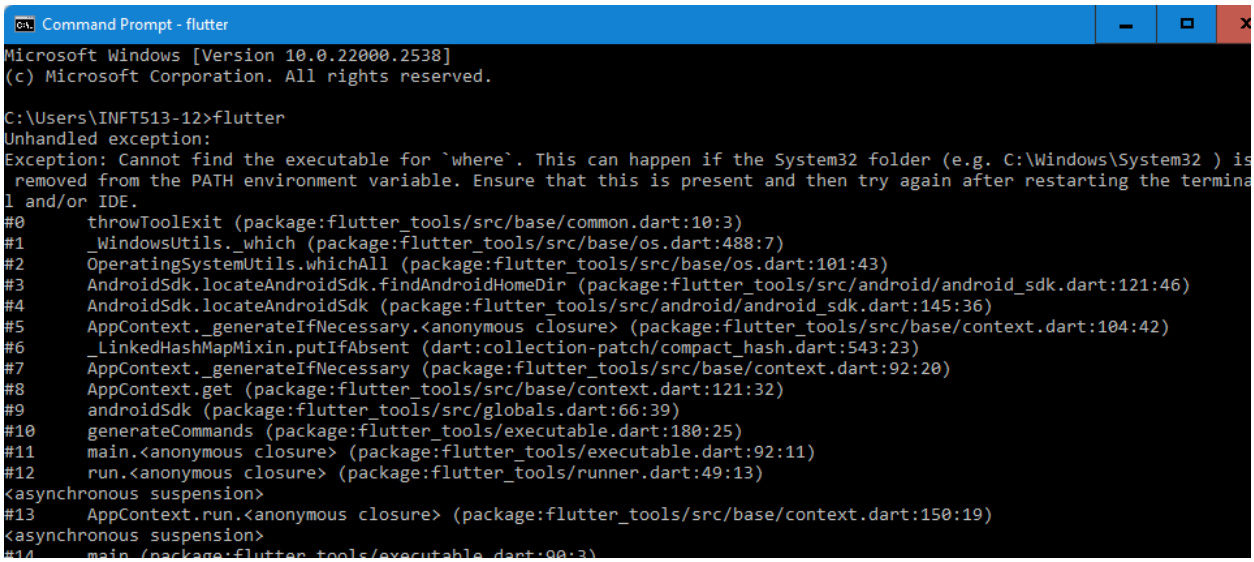


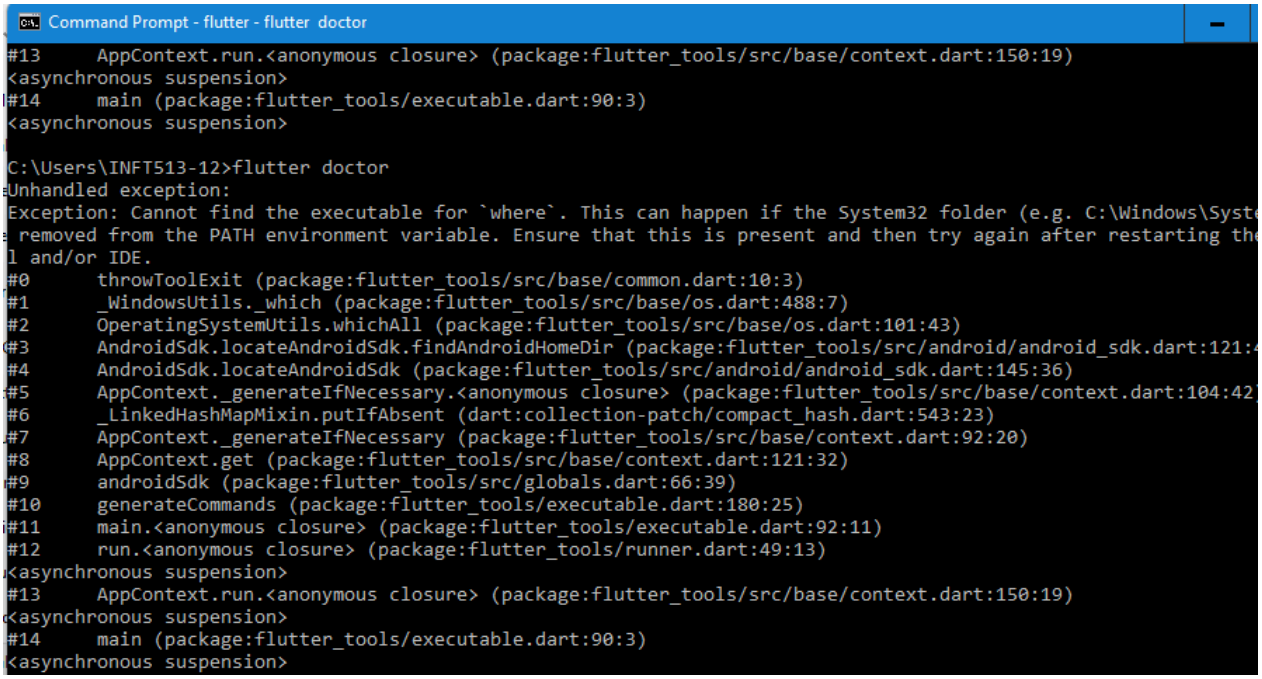
Step 4.2: Now, select path -> click on edit.

Step 4.3: In the above window, click on New ->write path of Flutter bin folder in variable value >ok ->ok ->ok



Step 5: Now, run the $ flutter command in command prompt.
Now, run the $ flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

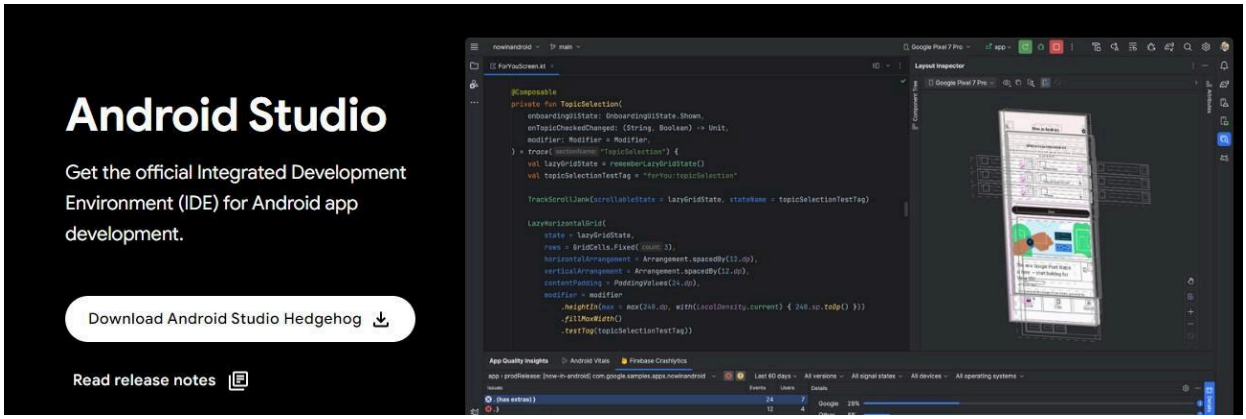




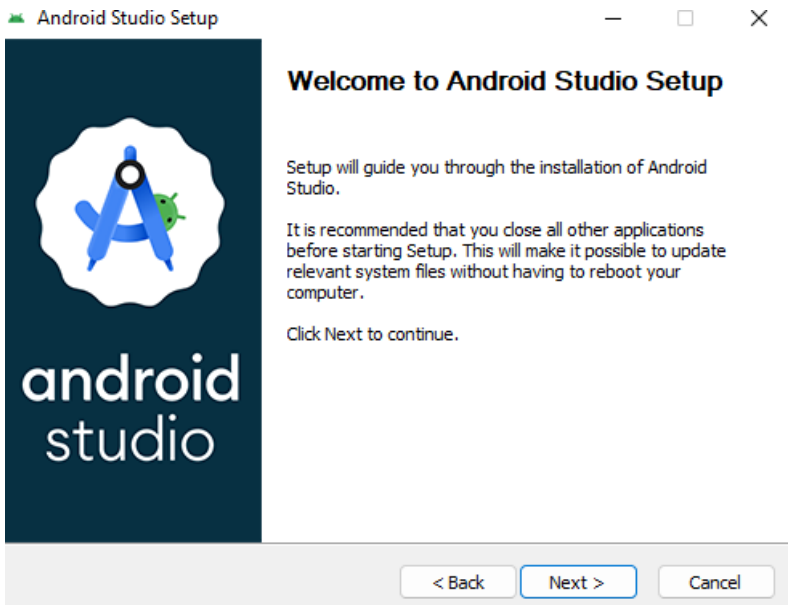
Step 6: When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which required to run Flutter as well as the development tools that are available but not connected with the Device.

Step 7: Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

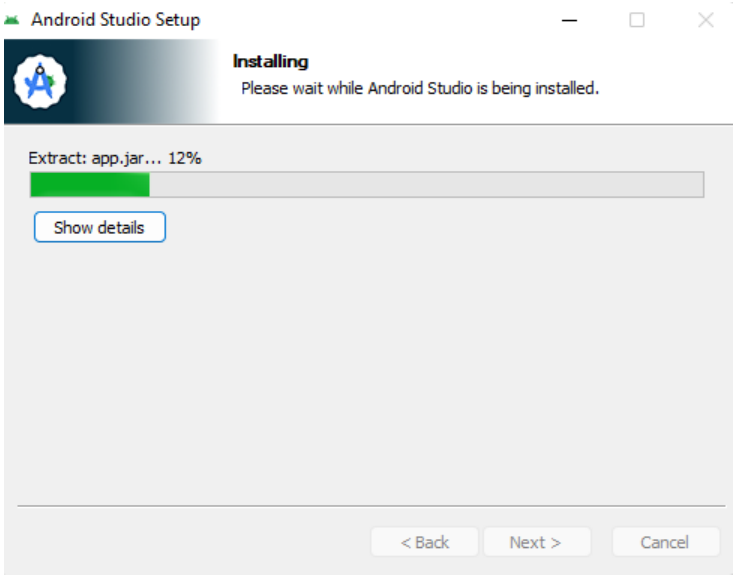
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



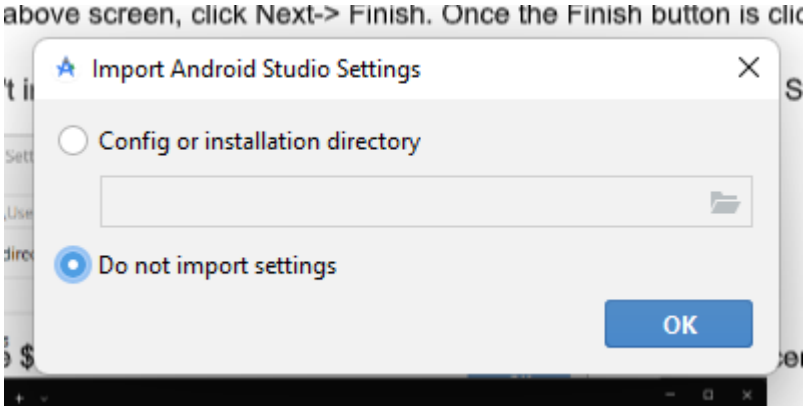
Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box.



Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the don't import Settings option' and click OK. It will start the Android Studio.



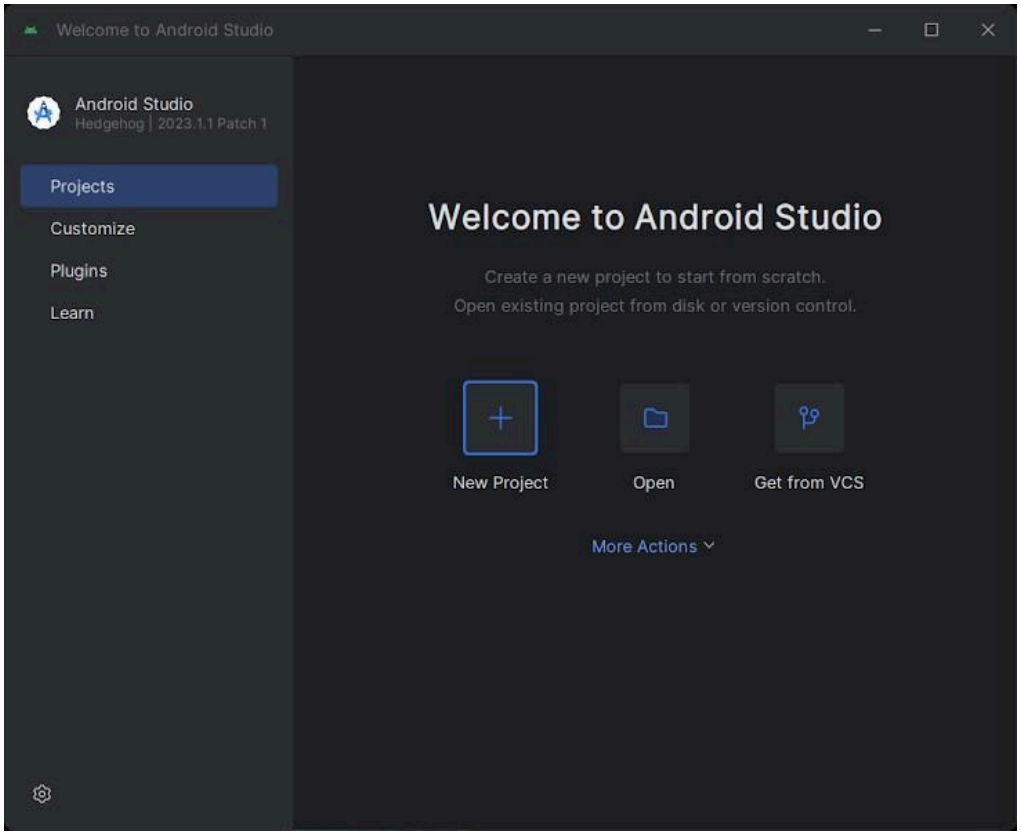
Step 7.5: run the $ flutter doctor command and Run flutter doctor --android-licenses command.

```
C:\Users\INFT513-12>flutter
Manage your Flutter app development.

Common commands:

  flutter create <output directory>
    Create a new Flutter project in the specified directory.

  flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                  Print this usage information.
-v, --verbose               Noisy logging, including all shell commands executed.
                            If used with "--help", shows hidden options. If used with "flutter doctor", shows additi
                            diagnostic information. (Use "-vv" to force verbose logging in those cases.)
-d, --device-id             Target device id or name (prefixes allowed).
    --version               Reports the version of this tool.
    --enable-analytics      Enable telemetry reporting each time a flutter or dart command runs.
    --disable-analytics     Disable telemetry reporting each time a flutter or dart command runs, until it is
                            re-enabled.
    --suppress-analytics    Suppress analytics reporting for the current CLI invocation.
```

```
C:\Users\INFT513-12>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[🖼] Flutter (the doctor check crashed)
    X Due to an error, the doctor check did not complete. If the error message below is not helpful
      about this issue at https://github.com/flutter/flutter/issues.
    X Exception: Cannot find the executable for `where`. This can happen if the System32 folder (e.
      C:\Windows\System32 ) is removed from the PATH environment variable. Ensure that this is pres
      again after restarting the terminal and/or IDE.
[√] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
    X cmdline-tools component is missing
      Run `path/to/sdkmanager --install "cmdline-tools;latest"`
      See https://developer.android.com/studio/command-line for more details.
[√] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Community 2019 16.11.17)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for
      development with C++" workload, and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
         - If there are multiple build tool versions available, install the latest
```

Step 8: Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.

Step 9: Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself. Do the following steps to install these plugins.



Conclusion:
the successful configuration of a comprehensive Flutter development environment, involving SDK download, Android emulator setup, and plugin integration, lays the foundation for efficient cross-platform app development. The meticulous setup process ensures a seamless and enjoyable experience, culminating in a well-prepared Flutter workspace for future experimentation and application development in the lab.