

Experiment No.1

MAD & PWA LAB

- **Aim:** Installation and Configuration of Flutter Environment.

- **Theory:**

Flutter is an opensource framework developed and supported by Google. Frontend and full-stack developers use Flutter to build an application's user interface (UI) for multiple platforms with a single codebase.

When Flutter launched in 2018, it mainly supported mobile app development. Flutter now supports application development on six platforms: iOS, Android, the web, Windows, MacOS, and Linux.

Flutter simplifies the process of creating consistent, appealing UIs for an application across the six platforms it supports. Because Flutter is a cross-platform development framework, we'll first compare it to native development. Then, we can highlight features that are unique to Flutter.

→ **The Advantages Of Flutter –**

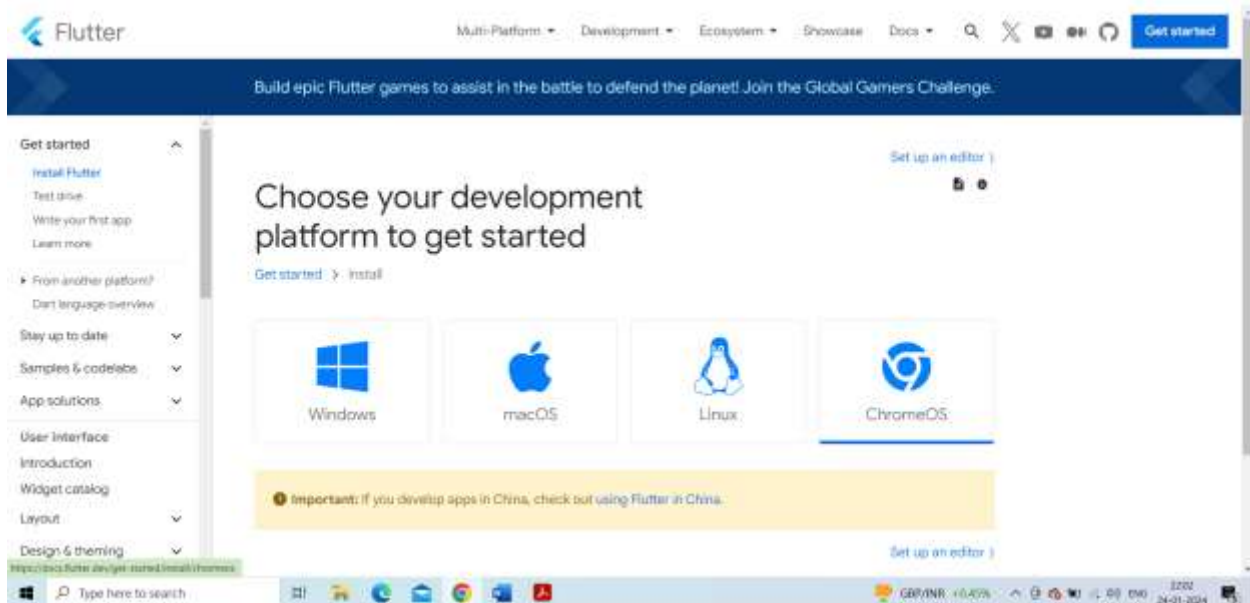
Here are some ways that Flutter stands out as a cross-platform development framework:

- Close-to-native performance. Flutter uses the programming language Dart and compiles into machine code. Host devices understand this code, which ensures a fast and effective performance.
- Fast, consistent, and customizable rendering. Instead of relying on platform-specific rendering tools, Flutter uses Google's open-source Skia graphic library to render UI. This provides users with consistent visuals no matter what platform they use to access an application.
- Developer-friendly tools. Google built Flutter with an emphasis on ease-of-use. With tools like hot reload, developers can preview what code changes will look like without losing state. Other tools like the widget inspector make it easy to visualize and solve issues with UI layouts.

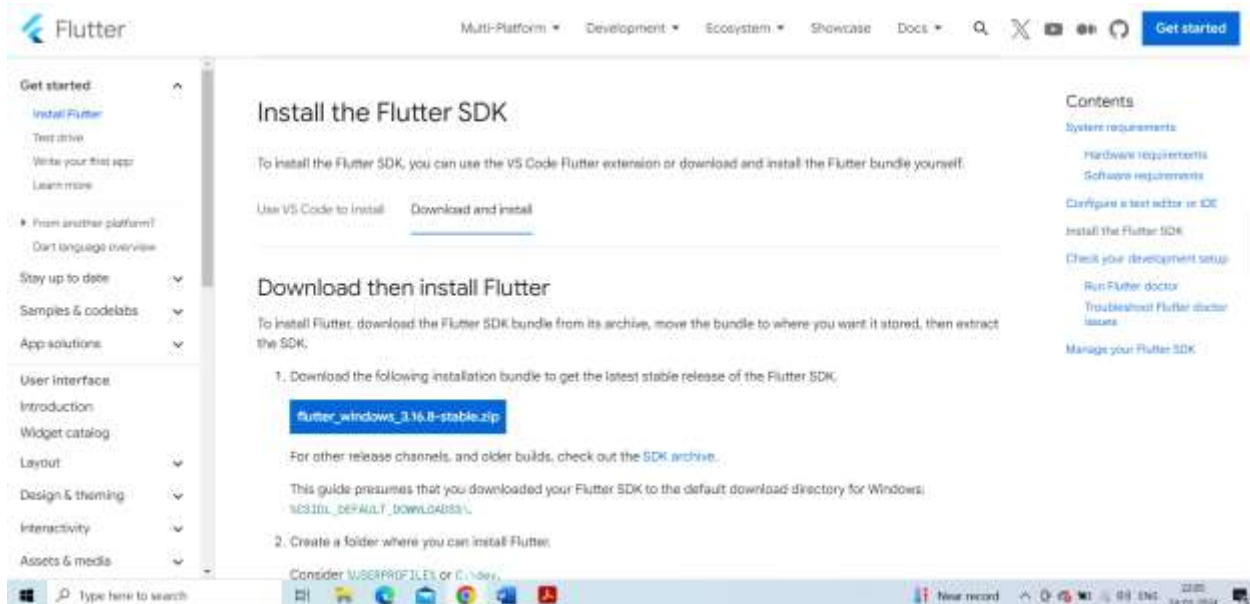


- **Install the Flutter SDK:**

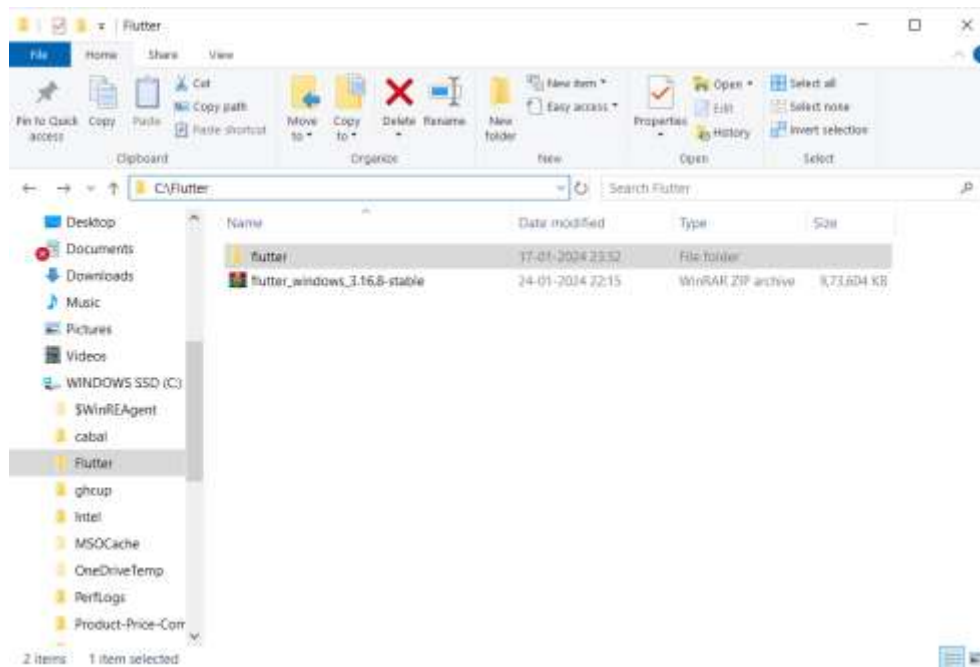
Step 1: Visit the official Flutter Software Development Kit website at <https://docs.flutter.dev/get-started/install> to access the installation bundle for Windows. Once on the website, you'll encounter the following screen.



Step 2: To obtain the most recent Flutter SDK, simply click on the Windows icon, where you'll discover the SDK download link.

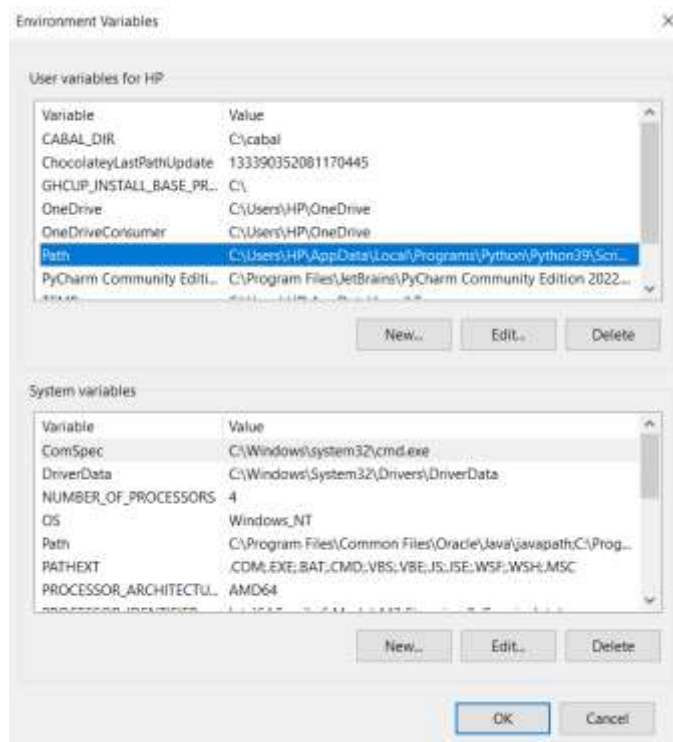


Step 3: After downloading, unzip the file and place it in your preferred installation folder or location, such as C:/Flutter.

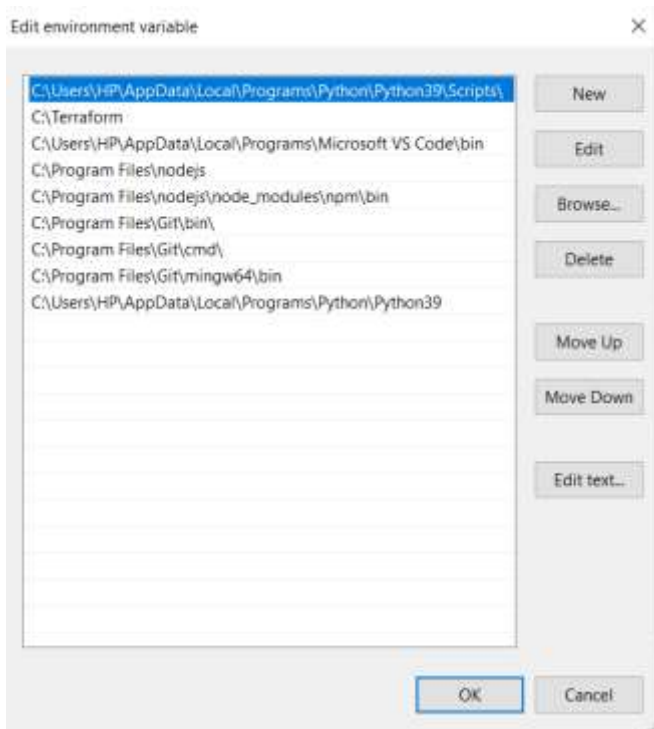


Step 4: Updating the system path on a regular Windows console to include the Flutter bin directory is necessary for running Flutter commands. To achieve this, follow the steps outlined below:

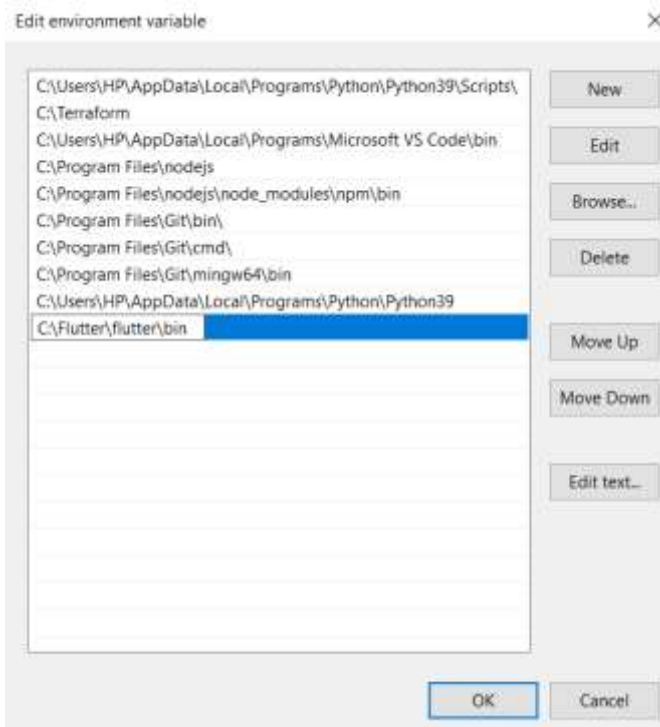
Step 4.1: Navigate to the properties of My Computer, then go to the advanced tab, and finally, access the environment variables. This will lead you to the corresponding screen.



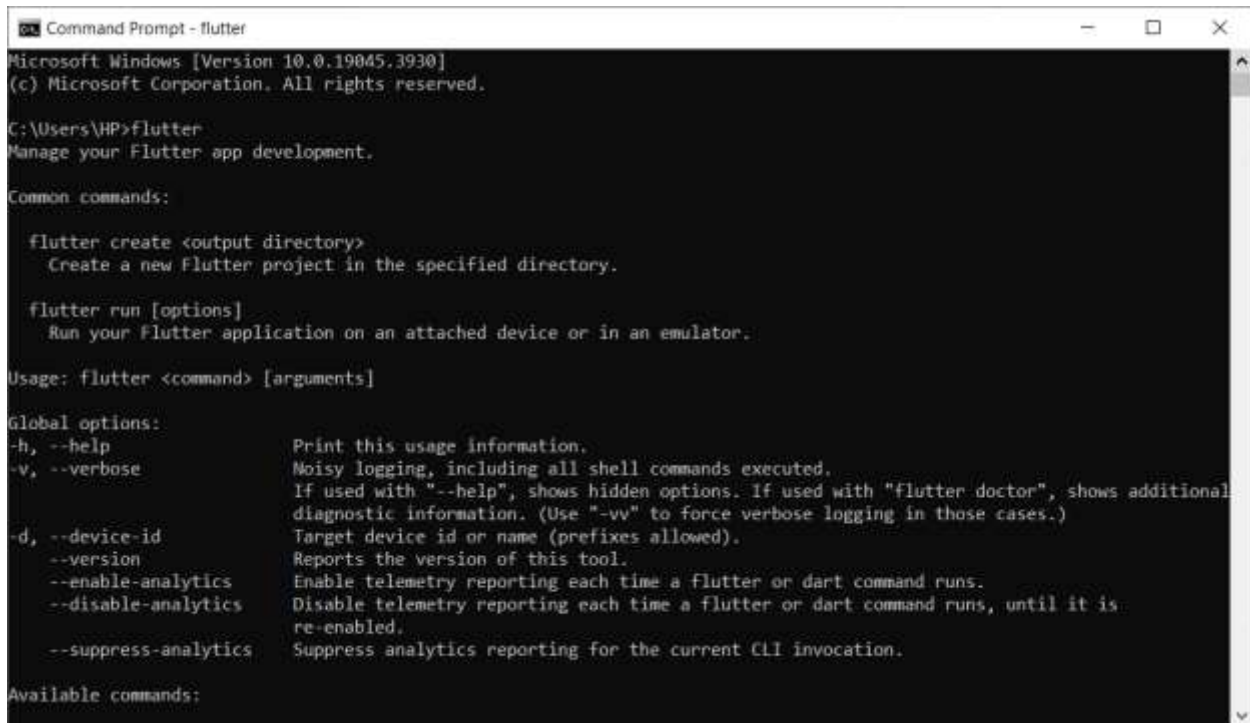
Step 4.2: Choose the "Edit" option after selecting the path, and you will be directed to the ensuing screen.



Step 4.3: Navigate to the New option in the current window, then input the path of the Flutter bin folder in the Variable Value field. Afterward, proceed to click OK, followed by additional OK prompts until you exit the window.



Step 5: Execute the command `$ flutter` in the command prompt.



```
Command Prompt - flutter
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>Flutter
Manage your Flutter app development.

Common commands:

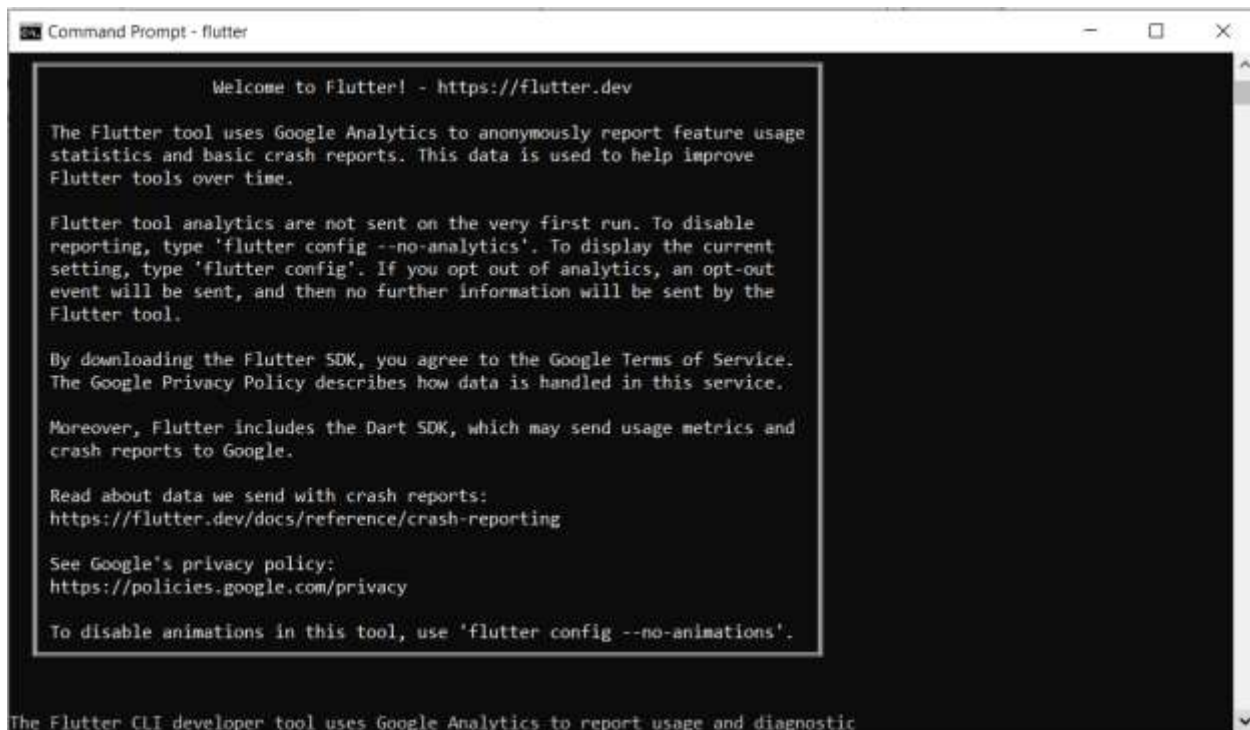
flutter create <output directory>
    Create a new Flutter project in the specified directory.

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 additional
                           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.

Available commands:
```



```
Command Prompt - flutter

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

To disable animations in this tool, use 'flutter config --no-animations'.

The Flutter CLI developer tool uses Google Analytics to report usage and diagnostic
```

Execute the "flutter doctor" command to assess the status of your Flutter installation by checking for all the necessary requirements for Flutter app development. This command generates a comprehensive report outlining the current state of your Flutter setup.

```
C:\Users\HP>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.8, on Microsoft Windows [Version 10.0.19045.3930], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.30)
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.85.2)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
```

Step 6: Upon executing the given command, a system analysis will be conducted, presenting a comprehensive report. This report will include information on any absent tools necessary for Flutter operation, along with details on available development tools that may not be linked to the device, as illustrated in the accompanying image.

```
C:\Users\HP>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.8, on Microsoft Windows [Version 10.0.19045.3930], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.30)
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.85.2)
[✓] Connected device (3 available)
[✓] Network resources

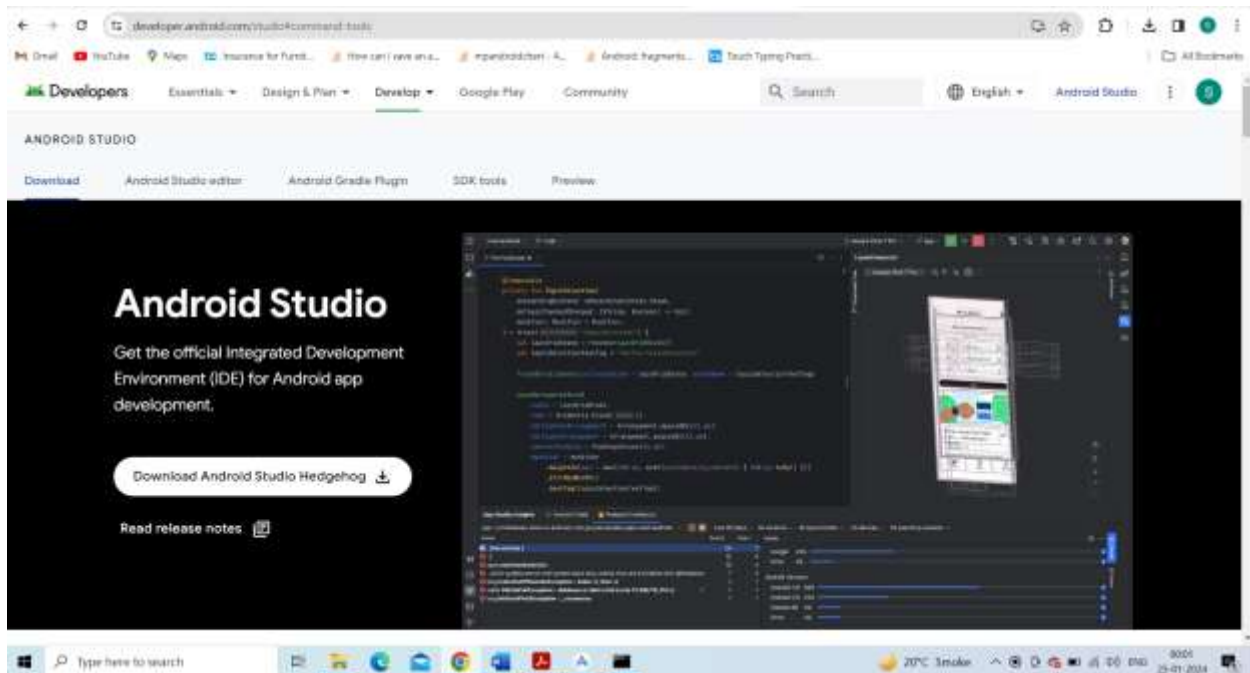
• No issues found!
```

In my laptop after executing the \$ flutter doctor command everything was fine and well connected there were no issues found.

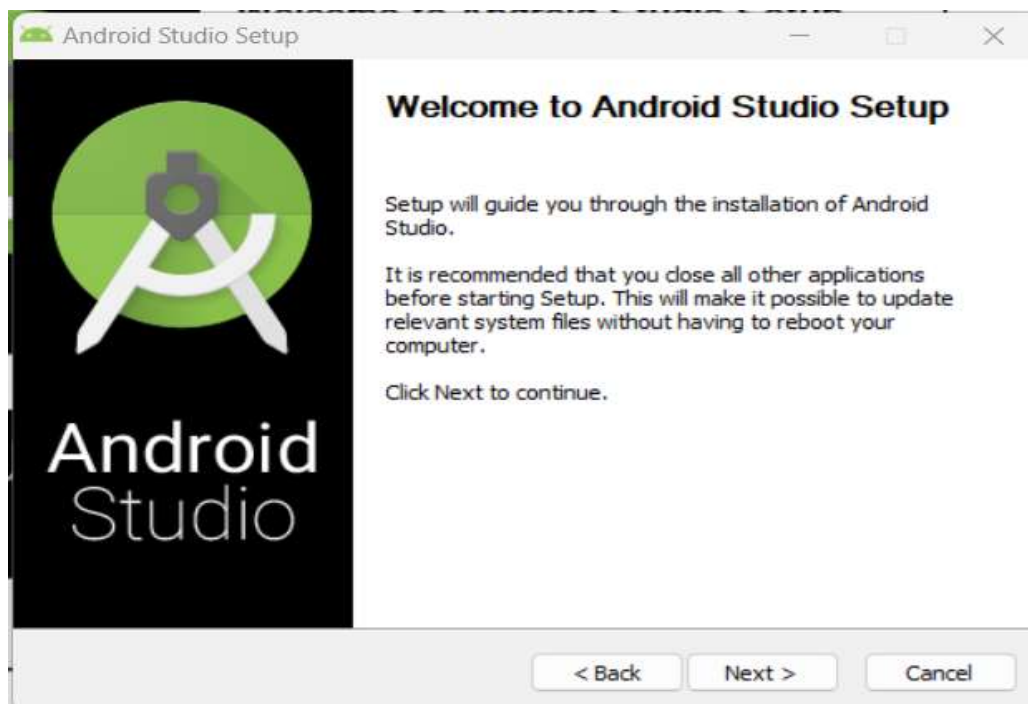
Step 7: If the flutter doctor command cannot locate the Android SDK tool on your system, you should initially install the Android Studio IDE. Follow these steps to install the Android Studio IDE.

[Initially Android Studio IDE was previously installed on my laptop, therefore there are no error messages displayed. However, if it is not installed, the subsequent steps are provided.]

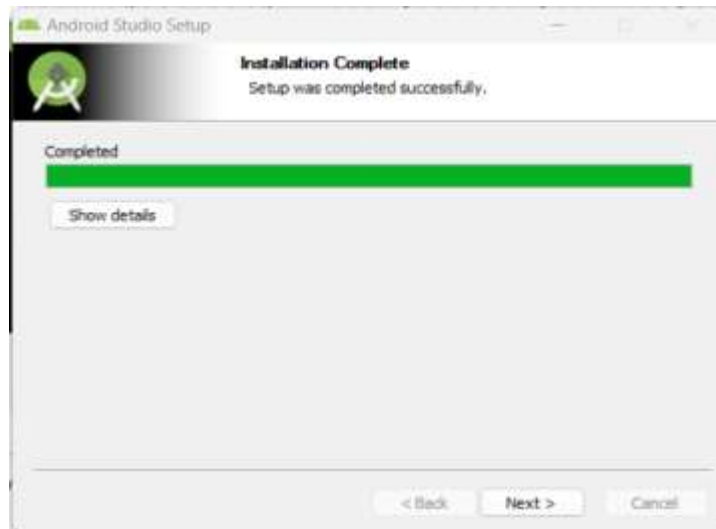
Step 7.1: Retrieve the most recent Android Studio executable or zip file directly from the official website.



Step 7.2: Upon completion of the download, proceed to open and run the .exe file. A dialog box will then appear as follows.



Step 7.3: Proceed with the installation wizard by following its steps. Once the wizard concludes, you will be presented with the following screen.



Step 7.4: Execute the command `$ flutter doctor` and then proceed to run `$ flutter doctor --android-licenses`.

```
C:\Users\HP>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.8, on Microsoft Windows [Version 10.0.19045.3930], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.2)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.30)
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.85.2)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
```

```
Command Prompt - flutter doctor --android-licenses - flutter doctor

y technical data, process, product, or service, directly or indirectly, to any country for which the United States govern-
ment or any agency thereof or the foreign government from where it is shipping requires an export license, or other gov-
ernmental approval, without first obtaining such license or approval. Recipient also agrees to implement measures to ens-
ure that foreign national employees are authorized to receive any information controlled by U.S. export control laws. An-
export is "deemed" to take place when information is released to a foreign national wherever located.

30.7 Special Terms for Pre-Release Materials: If so indicated in the description of the Evaluation Software, the Evaluat-
ion Software may contain Pre-Release Materials. Recipient hereby understands, acknowledges and agrees that: (i) Pre-Rele-
ase Materials may not be fully tested and may contain bugs or errors; (ii) Pre-Release materials are not suitable for co-
mercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or FCC) have n-
ot been obtained, and Pre-Release Materials may therefore not be certified for use in certain countries or environments;
or may not be suitable for certain applications and (iv) MIPS can provide no assurance that it will ever produce or make
generally available a production version of the Pre-Release Materials. MIPS is not under any obligation to develop and
/or release or offer for sale or license a final product based upon the Pre-Release Materials and may unilaterally elect
to abandon the Pre-Release Materials or any such development platform at any time and without any obligation or liabil-
ity whatsoever to Recipient or any other person.

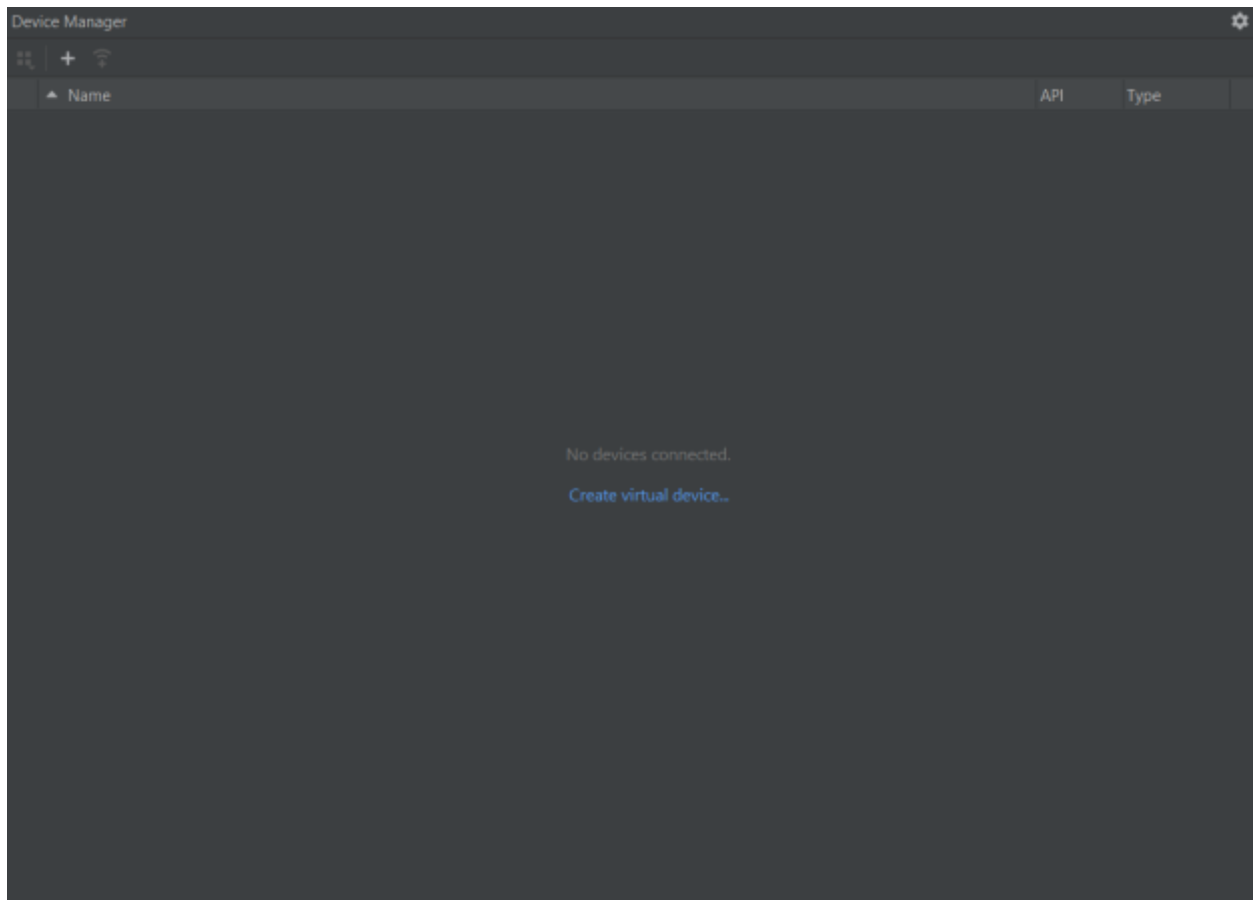
ANY PRE-RELEASE MATERIALS ARE NON-QUALIFIED AND, AS SUCH, ARE PROVIDED AS IS AND AS AVAILABLE, POSSIBLY WITH FAULTS, AND
WITHOUT REPRESENTATION OR WARRANTY OF ANY KIND.

30.8 Open Source Software: In the event Open Source software is included with Evaluation Software, such Open Source soft-
ware is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source softwar-
e comments in the applicable source code file(s) and/or file header as indicated in the Evaluation Software. Additional
detail may be available (where applicable) in the accompanying on-line documentation. With respect to the Open Source so-
ftware, nothing in this Agreement limits any rights under, or grants rights that supersede, the terms of any applicable
Open Source software license agreement.

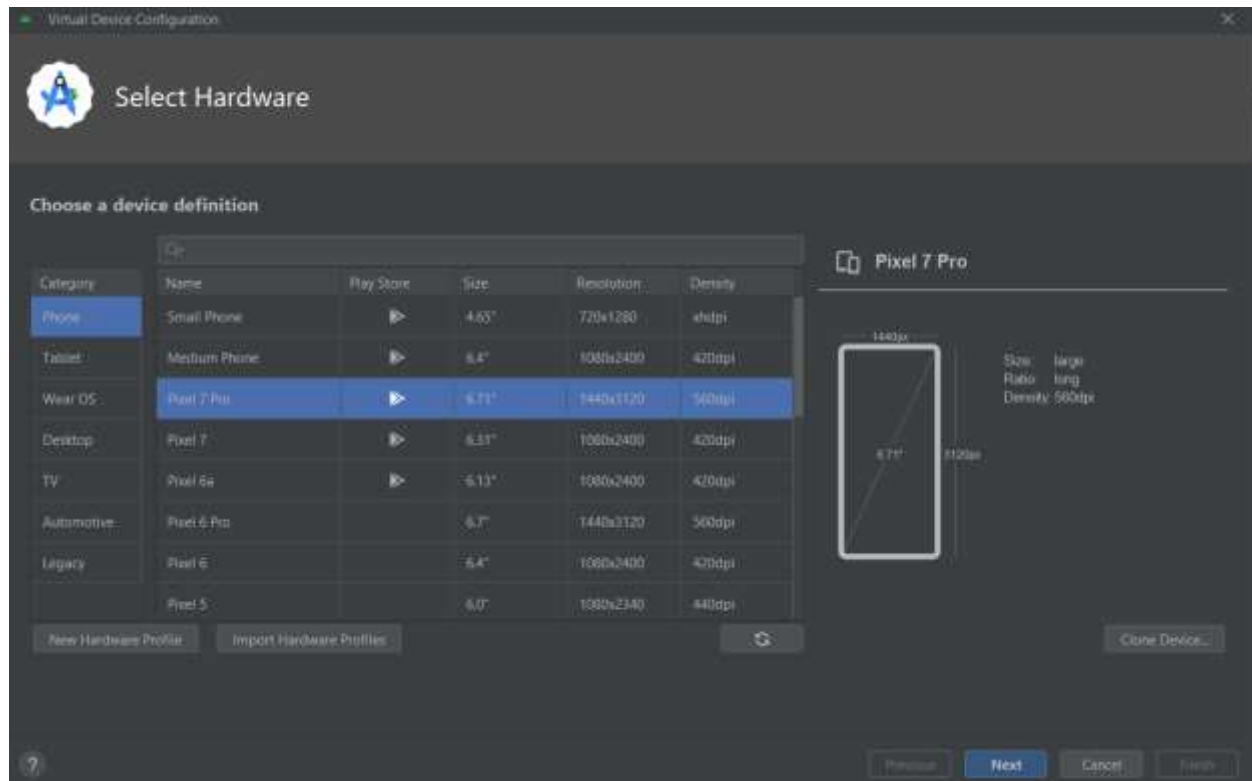
*****
Accept? (y/N): y
All SDK package licenses accepted
```


Step 8: Following that, you should configure an Android emulator, which is tasked with executing and testing the Flutter application.

Step 8.1: Create a virtual Android emulator by navigating to Android Studio, then selecting Tools > Android > AVD Manager. Alternatively, you can access it through Help -> Find Action, and type "Emulator" in the search box to bring up the corresponding screen.

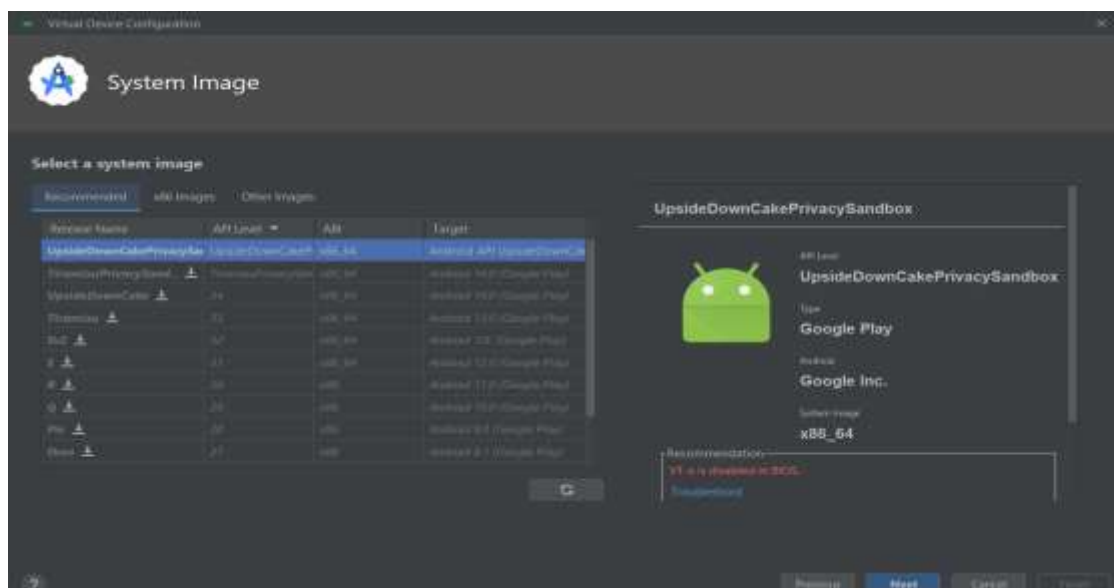


Step 8.2: Select the device definition and proceed by clicking on Next.



Step 8.3: Choose the system image corresponding to the most recent Android version, then proceed by clicking on Next.

Step 8.4: If the AVD configuration is accurate, proceed to click on Finish, and the subsequent screen will be displayed.

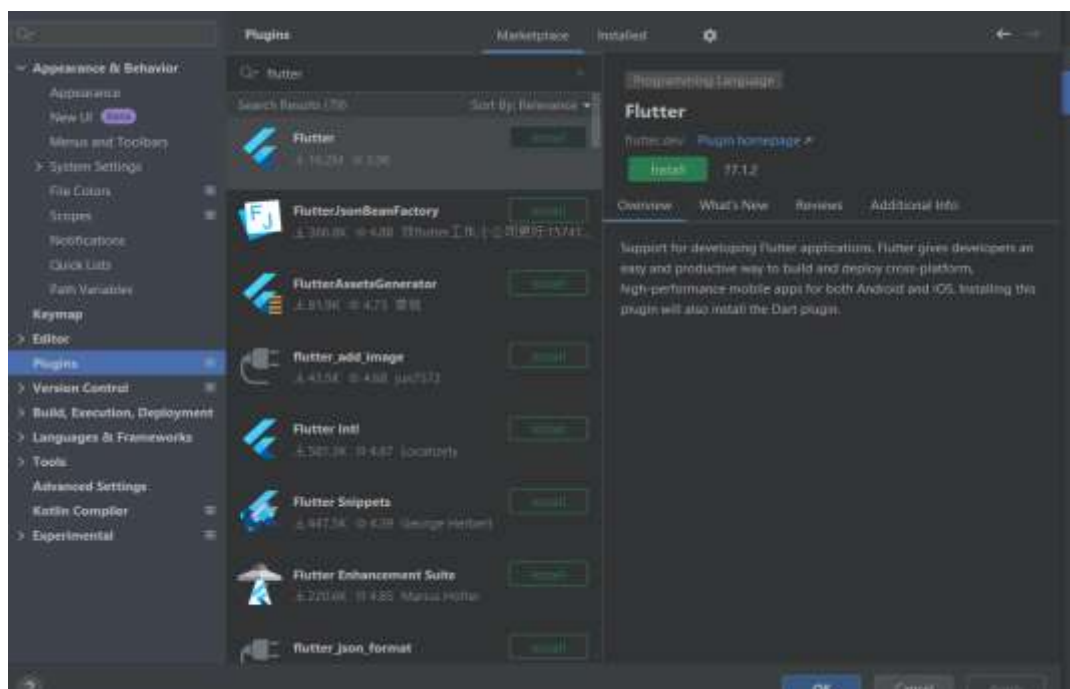


Step 8.5: After running the created virtual device, The Android emulator is displayed as below screen.

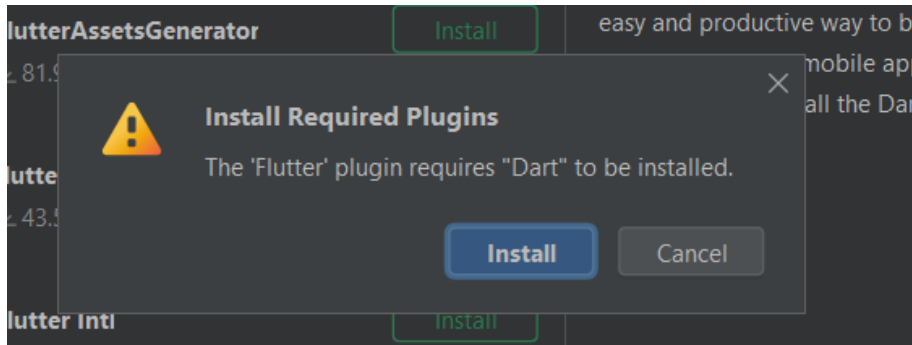


Step 9: To set up the Flutter and Dart plugins for developing Flutter applications in Android Studio, follow these steps. These plugins offer a template for creating Flutter applications and allow you to run and debug them directly within Android Studio.

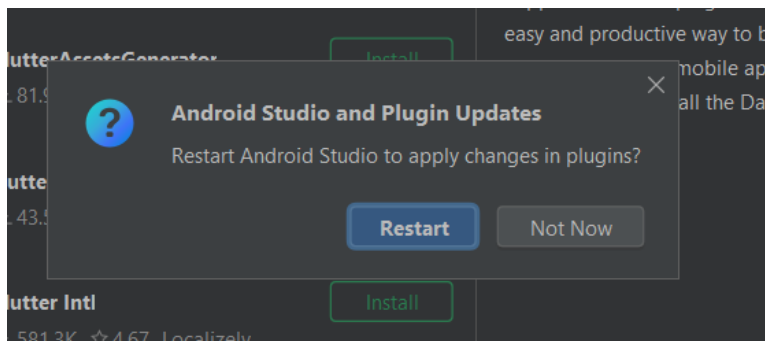
Step 9.1: Launch Android Studio, navigate to Settings by selecting File, then go to Plugins.



Step 9.2: Locate the Flutter plugin, and once found, proceed to install it. Upon initiating the installation, a prompt will appear asking you to install the Dart plugin. Confirm the installation by clicking on install.



Step 9.3: To implement the modifications made to the plugins, please restart Android Studio.



- **Conclusion:**

Therefore, we have gained an understanding of the process for installing and configuring the Flutter environment. This involves the installation of the Flutter SDK, setting up Android Studio, and ultimately creating and integrating a virtual device into Android Studio.