MAD PWA Lab

EXPERIMENT 1

Name : Pranathi Narsupalli
Div : D15B Roll no. : 45

AIM : Installation and configuration of flutter environment.

THEORY : Flutter is an open-source UI software development toolkit created by Google. It allows developers to build natively compiled applications for mobile, web, and desktop from a single codebase. Flutter uses the Dart programming language and provides a rich set of pre-designed widgets for creating modern and responsive user interfaces. It's known for its hot reload feature, which enables quick code changes and instant updates in the app, making the development process more efficient. Flutter comes with a variety of features that make it a popular choice for developing cross-platform applications. Here are some key features of Flutter:

1. Single Codebase: Developers can write code once and deploy it on multiple platforms, including iOS, Android, web, and desktop, reducing development time and effort.

2. Hot Reload: One of the standout features of Flutter, Hot Reload allows developers to see the results of code changes instantly in the running app, making the development process faster and more interactive.

3. Expressive UI: Flutter provides a rich set of customizable widgets, enabling developers to create expressive and visually appealing user interfaces that look consistent across different platforms.

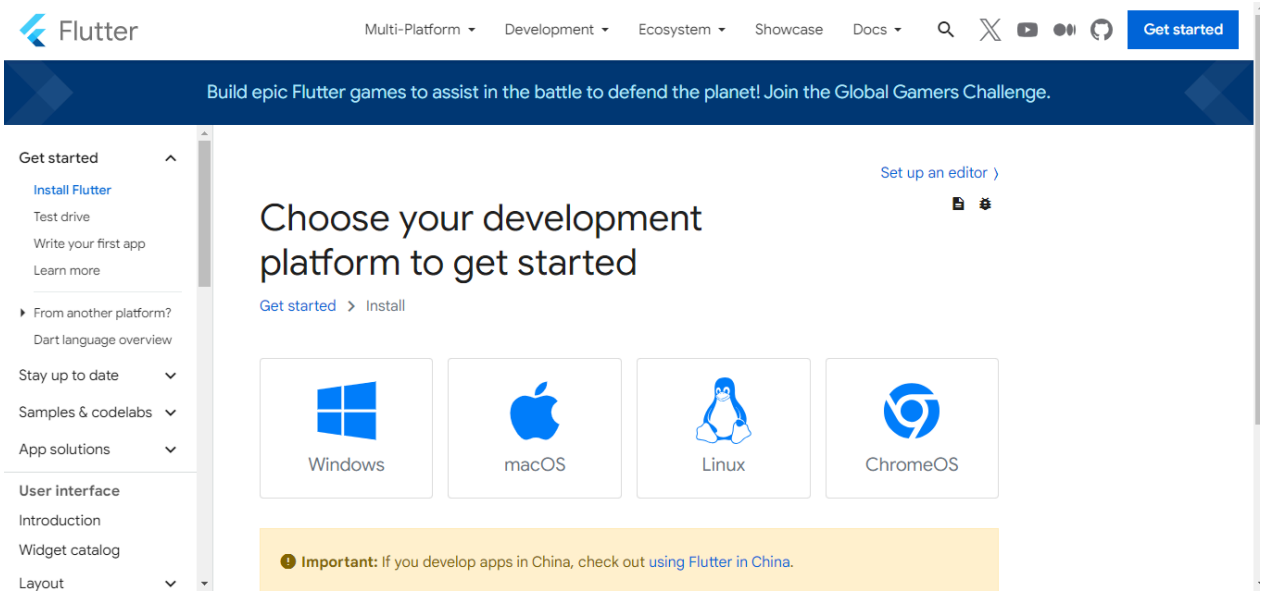
4. Fast Development: The combination of Hot Reload, a rich widget library, and a concise language (Dart) makes Flutter a fast and productive framework for app development.

5. High Performance: Flutter apps are compiled to native ARM code, providing high performance and a smooth user experience. The framework also leverages the Skia graphics engine for efficient rendering.

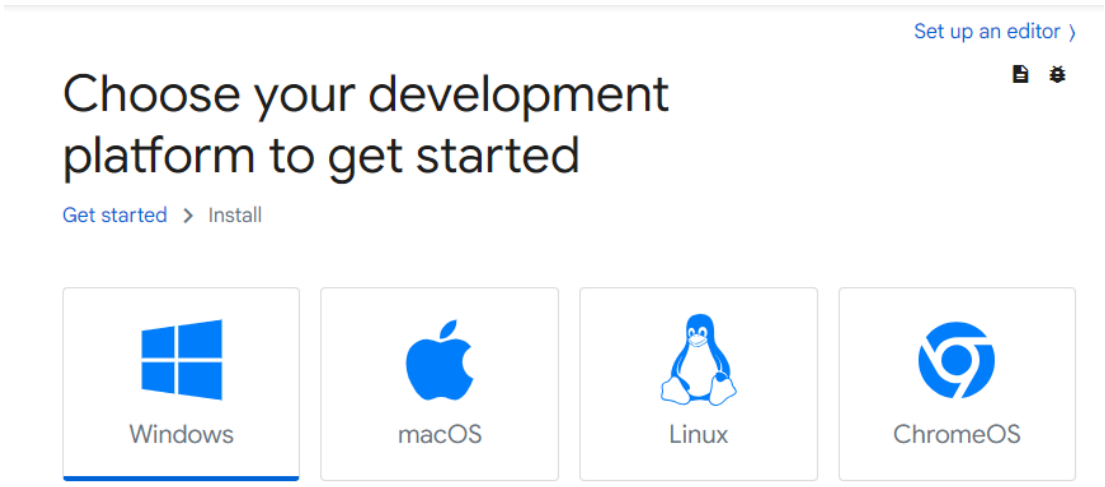
6. Adaptable Design: Flutter's flexible architecture enables developers to create adaptive designs that work well on various screen sizes and orientations.

Install the Flutter SDK

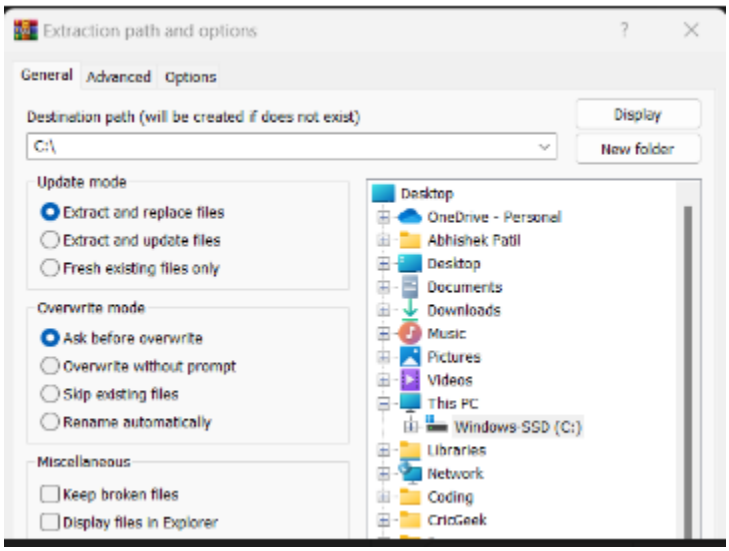
Step 1 : Download the installation bundle of the Flutter Software Development Kit for Windows from its official website https://docs.flutter.dev/get-started/install .



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

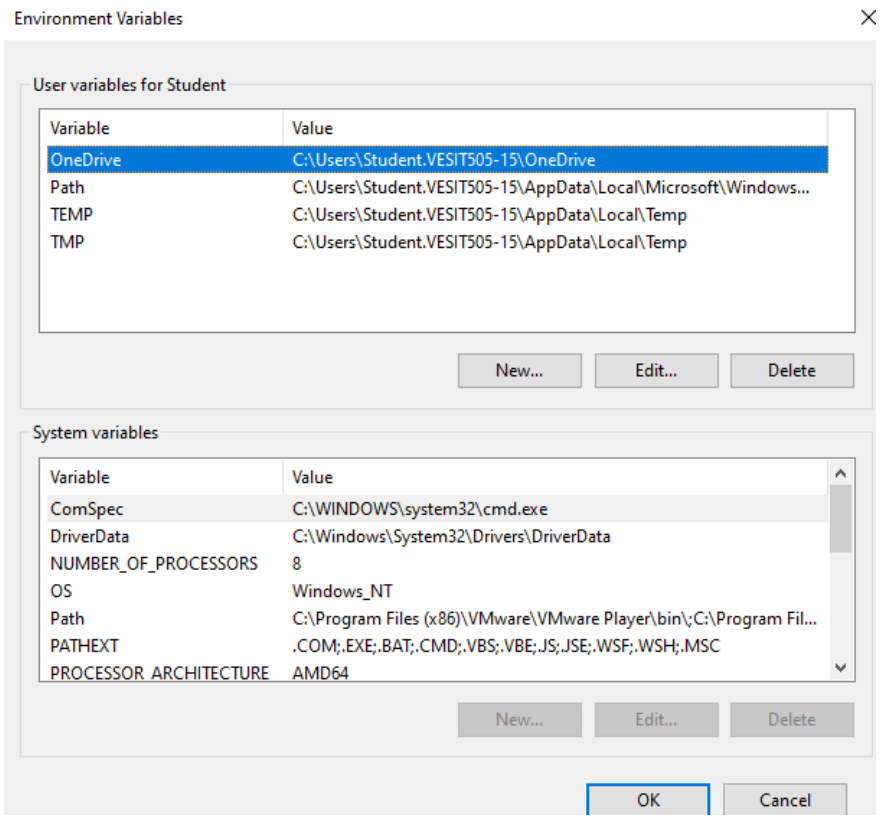


Step 3: When your download is complete, extract the zip file and place it in the desired installation folder or location, for example, C: /Flutter.

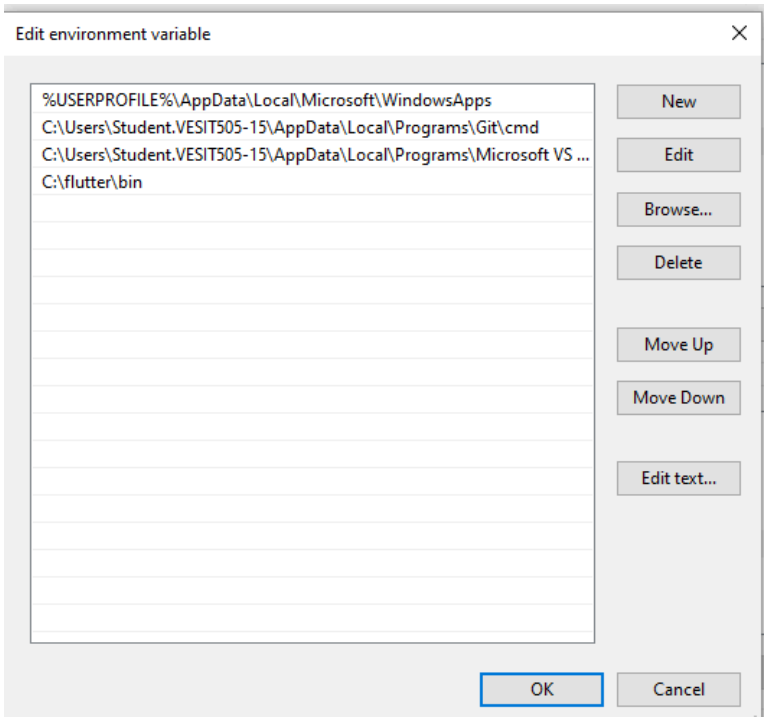


Step 4: To run the Flutter command in regular windows console, you need to update the system path to include the flutter bin directory. The following steps are required to do this:

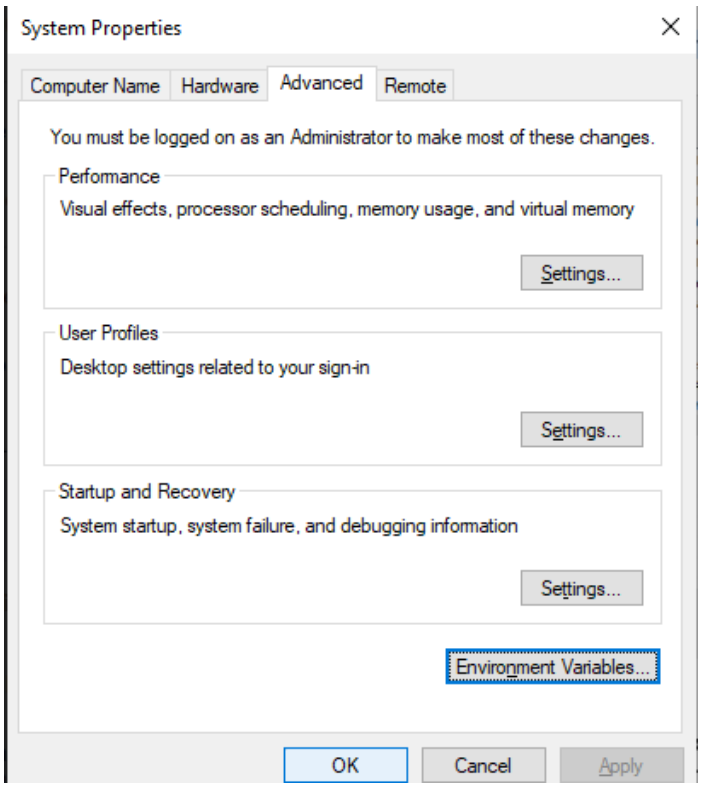
Step 4.1: Go to MyComputer properties -> advanced tab -> environment variables. You will get the following screen.



Step 4.2: Now, select path -> click on edit. The following screen appears

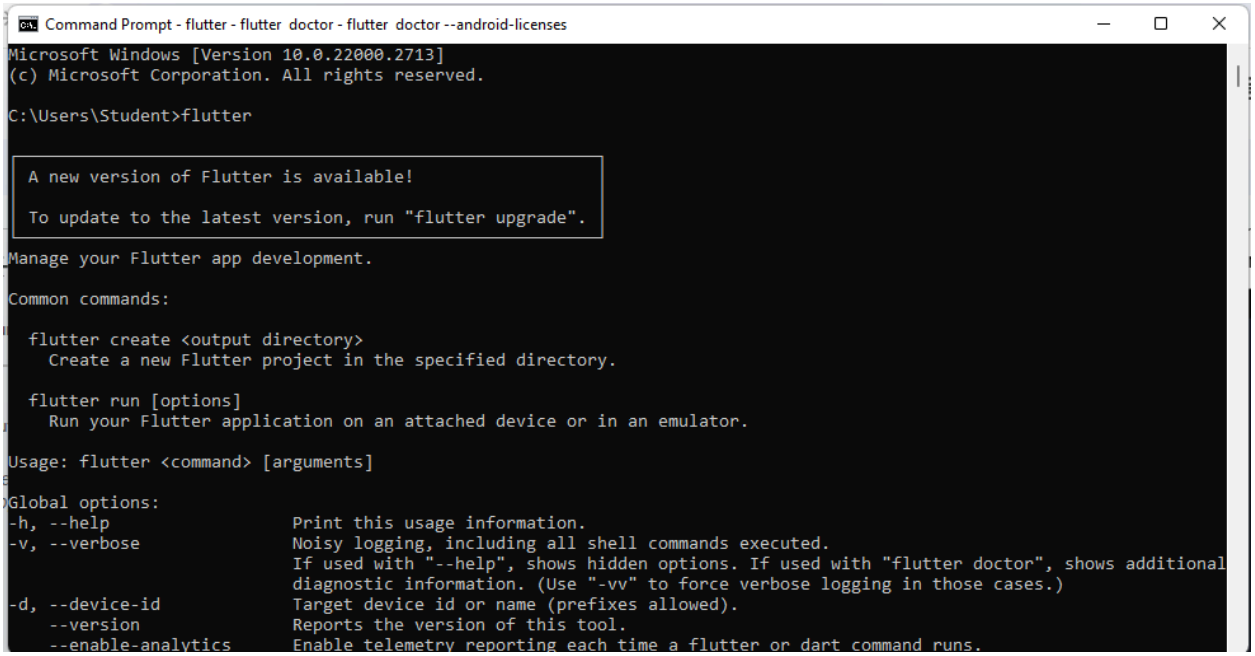


Step 4.3: In the above window, click on New -> ;write path of Flutter bin folder in variable value -> ok -> ok -> ok.

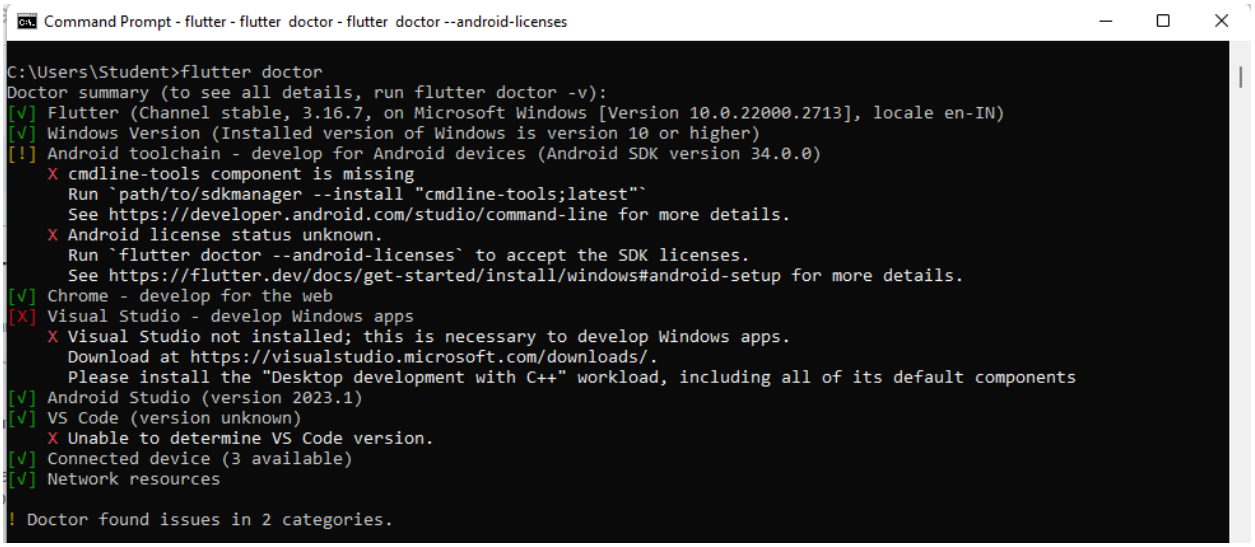


Step 5: Now, run the $ flutter command in command prompt.

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.



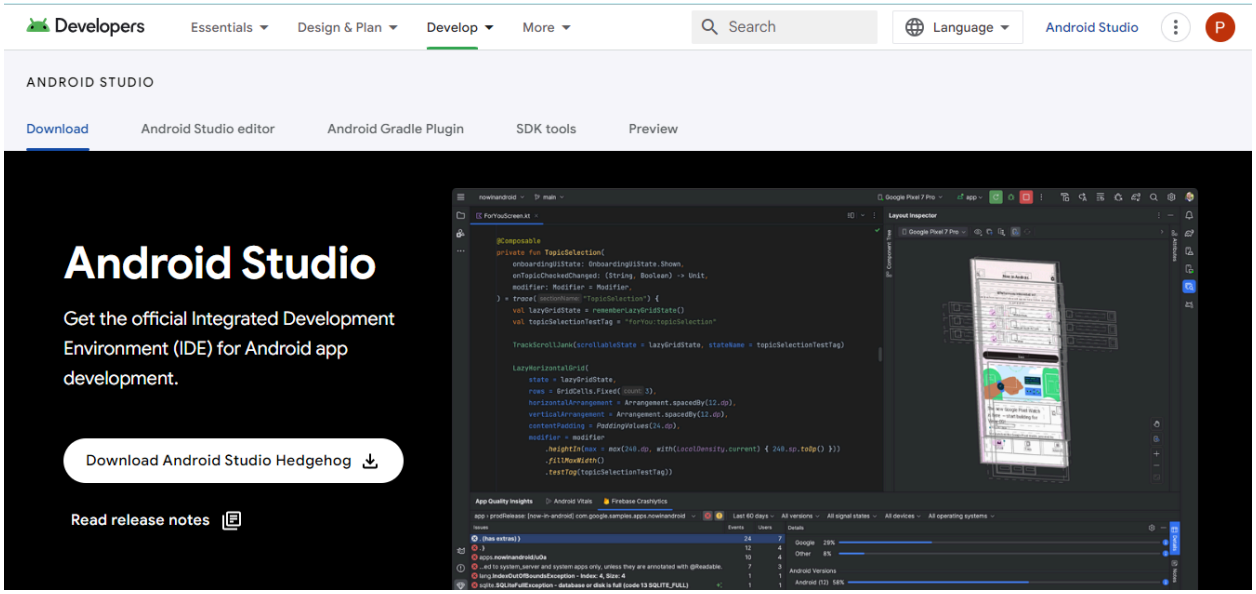
Step 6: When you run the above command, it will analyze the system and show its report, as shown in the below image. Here, you will find the details of all missing tools, which required to run Flutter as well as the development tools that are available but not connected with the device.



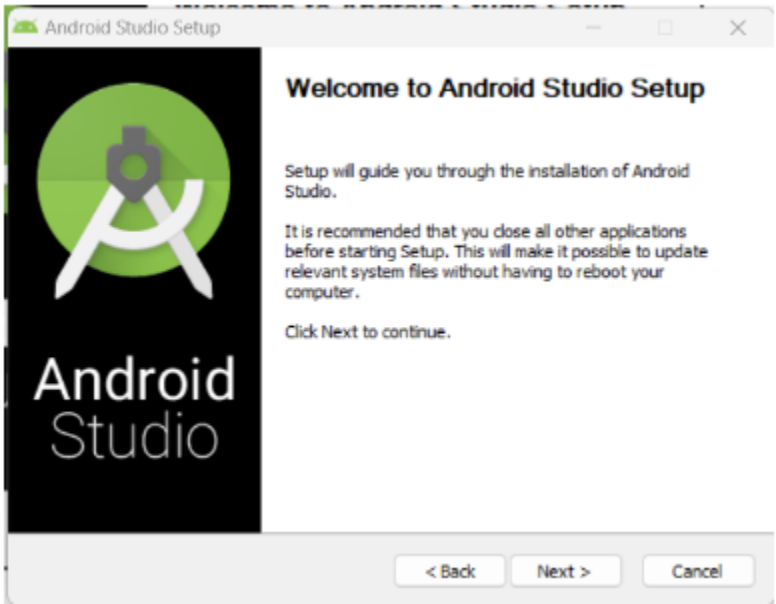
Step 7: Install the Android SDK. If the flutter doctor command does not find the Android SDK

tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

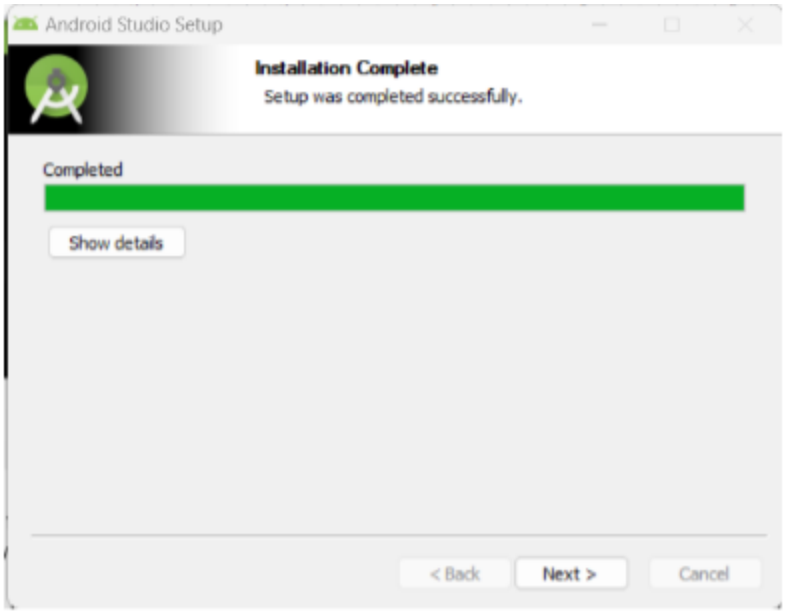
Step 7.1: Download the latest Android Studio executable or zip file from the official site.



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.



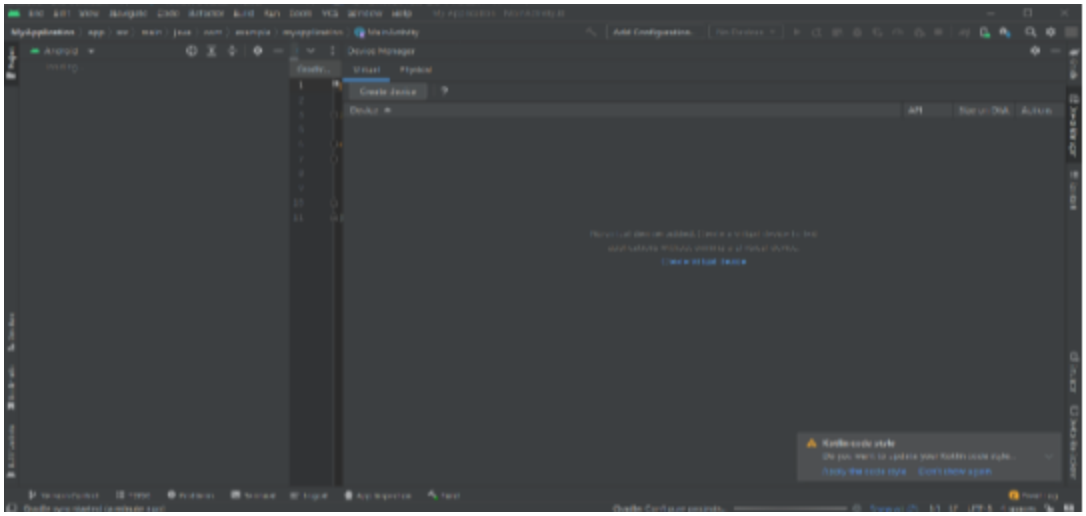
Step 7.4: In the above screen, click Next -> Finish. import Settings option' and click OK. It will start the Android Studio.

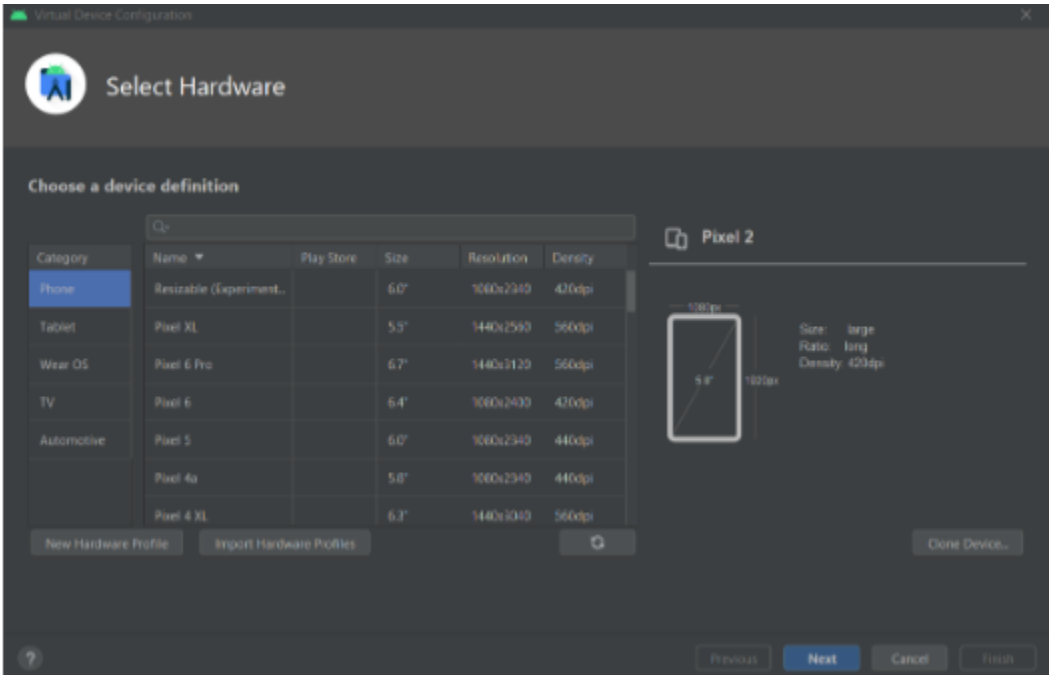
Step 7.5: run the $ flutter doctor command and Run flutter doctor --android-licenses command.

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

Step 8.1: To set an Android emulator, go to Android Studio -> Tools -> Android -> AVD Manager and select Create Virtual Device. Or, go to Help -> Find Action -> type Emulator in the search box. You will get the following screen.

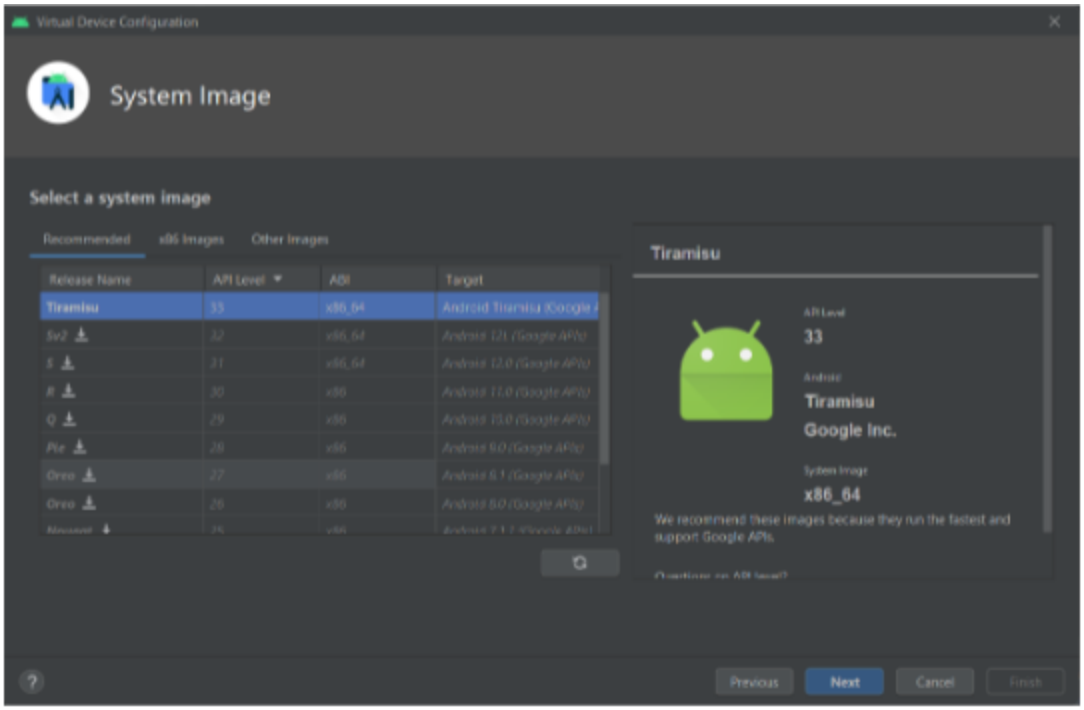
Step 8.2: Choose your device definition and click on Next.



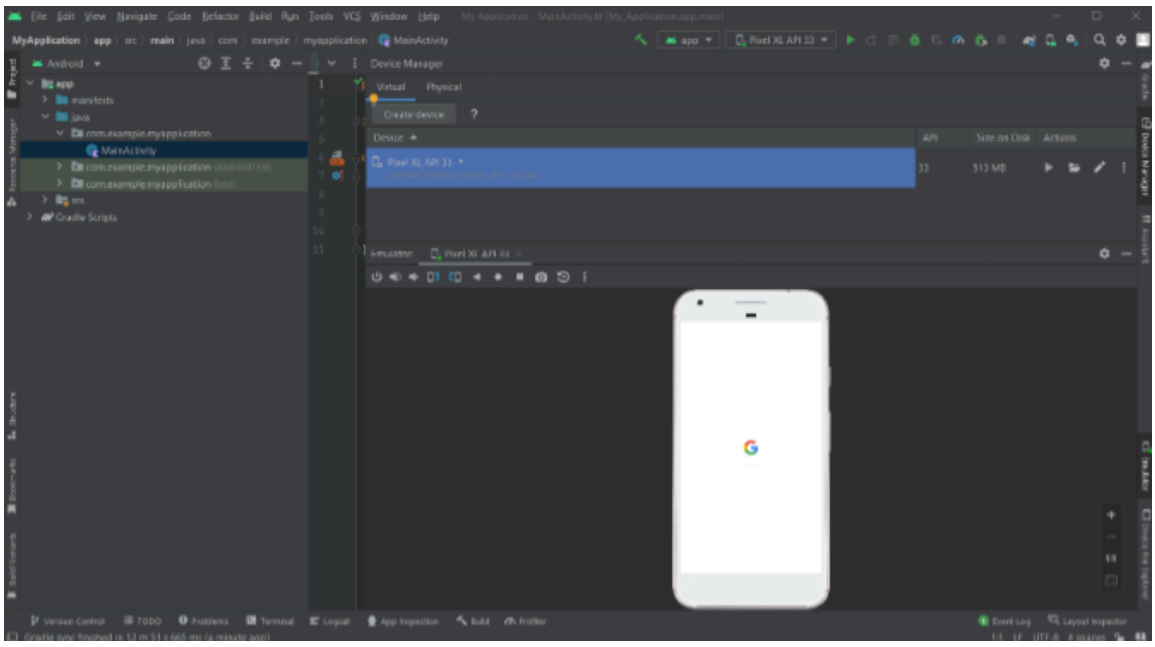


Step 8.3: Select the system image for the latest Android version and click on Next.

Step 8.4: Now, verify the all AVD configuration. If it is correct, click on Finish. The following screen appears.

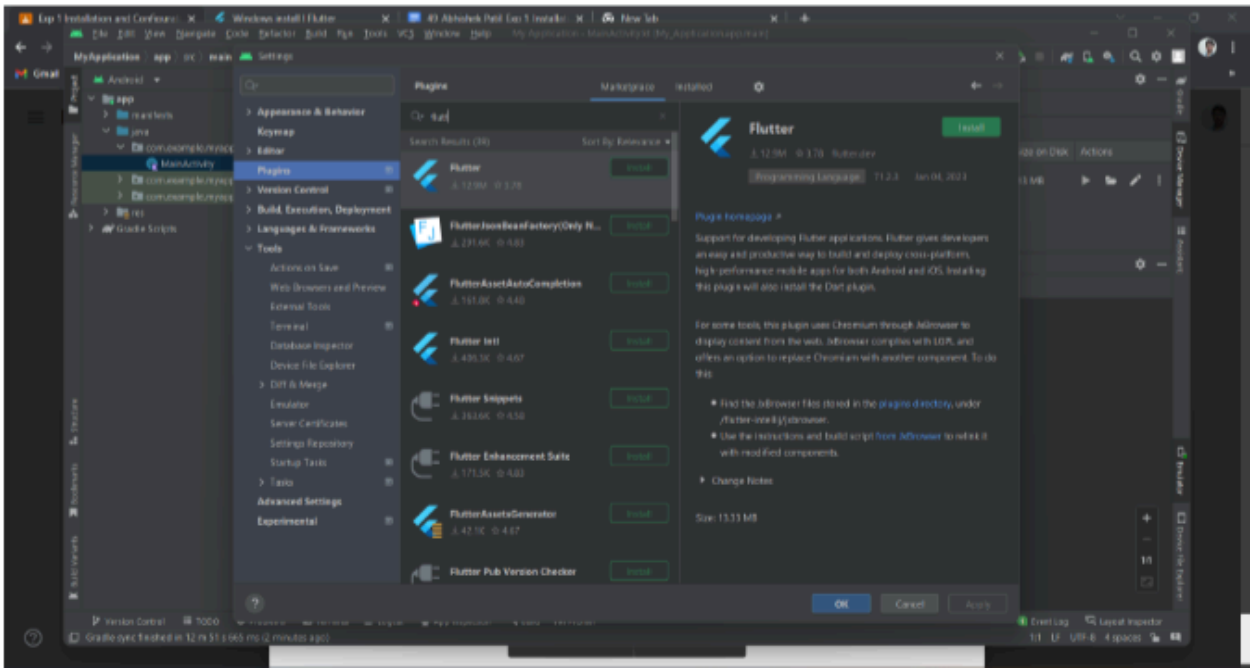


Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.



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.

Step 9.1: Open the Android Studio and then go to File -> Settings -> Plugins.



Step 9.2: Now, search the Flutter plugin. If found, select Flutter plugin and click install. When you click on install, it will ask you to install Dart plugin as below screen. Click yes to proceed.

Step 9.3: Restart the Android Studio.

CONCLUSION : We implemented the installation and configuration of the Flutter environment to enable developers to efficiently build cross-platform applications with a unified codebase. Setting up Flutter lets developers create apps for both iOS and Android using the same code. With handy features like Hot Reload for quick changes and access to device functions, it makes the development process faster. Flutter also supports a consistent and good-looking design across different devices, and integrating with Firebase adds useful backend services. Overall, Flutter simplifies cross-platform app development with an easy-to-use environment.