Experiment 1

Experiment No 1 1: Installation and Configuration of Flutter Environment.	
ROLL NO	**32**
NAME	**Priyank Kulkarni**
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPED	

Introduction:
Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications from a single codebase for any web browser, Fuchsia, Android, iOS, Linux, macOS, and Windows. First described in 2015, Flutter was released in May 2017.

The major components of Flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter Development Tools (DevTools)

Dart language

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.For better performance, release versions of Flutter apps on all platforms use ahead-of-time (AOT) compilation, except for on the Web, where code is transpiled to JavaScript.

Flutter inherits Dart's Pub package manager and software repository, which allows users to publish and use custom packages as well as Flutter-specific plugins.

Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using either Google's Skia graphics library or the custom "Impeller" graphics layer. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS to implement accessibility, file and network I/O, native plugin support, and more.

Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

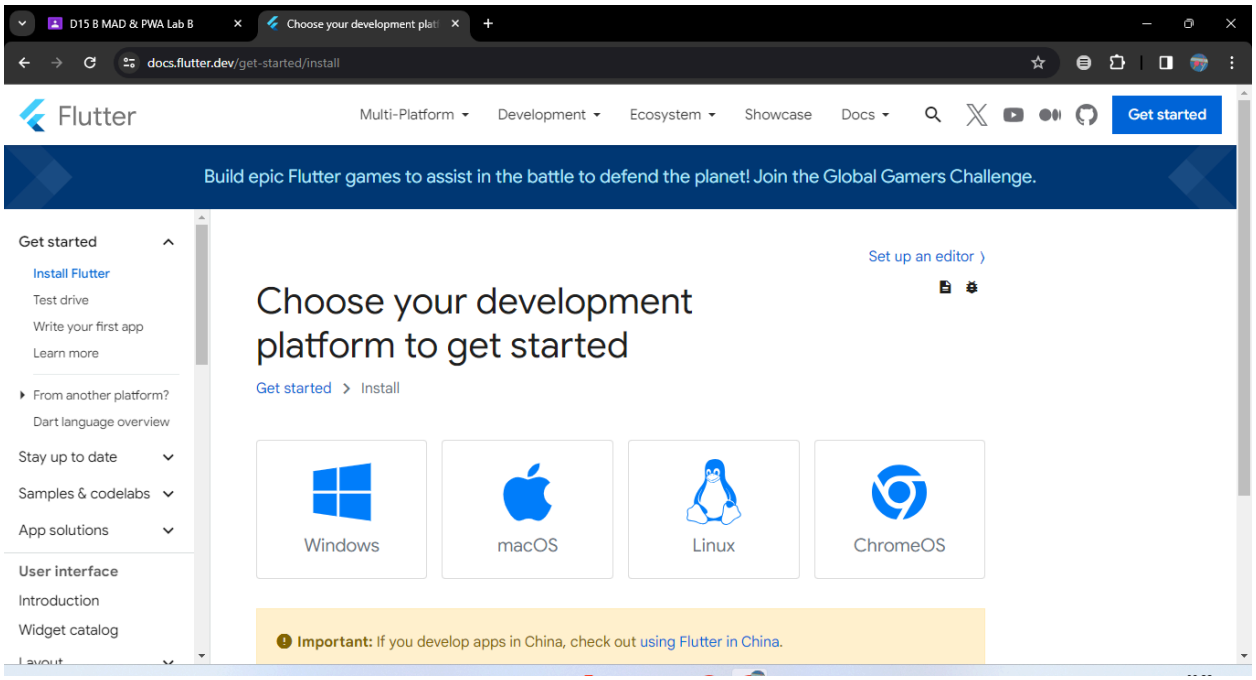
Design-specific widgets

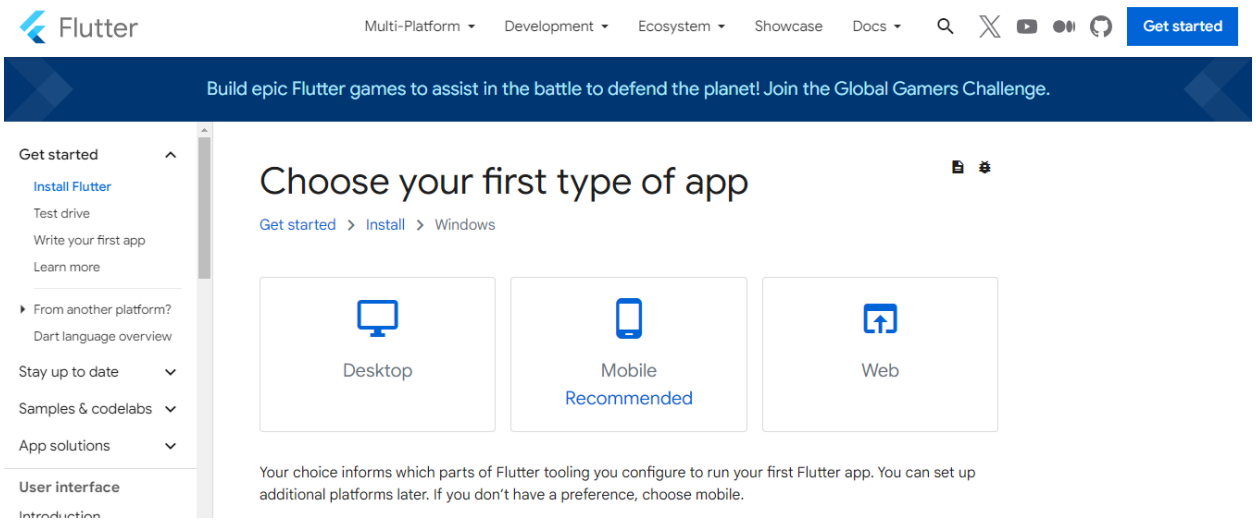
The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same

name, and *Cupertino* widgets implement Apple's iOS Human interface guidelines. Flutter allows the developer to use either set of widgets on either platform, i.e. even Cupertino widgets on Android. Third party packages can be used to automatically adjust the app's design to the current operating system.

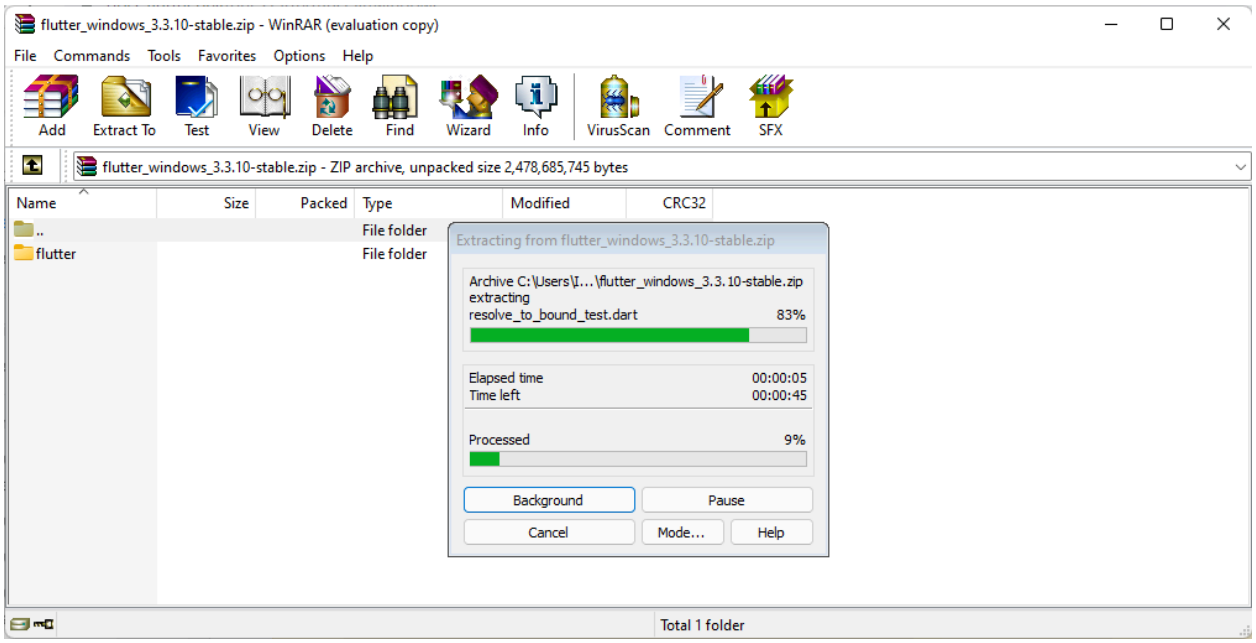
Flutter Installation:

Step 1: Install the Flutter SDK Download the Flutter Software Development Kit from the official website for Windows.





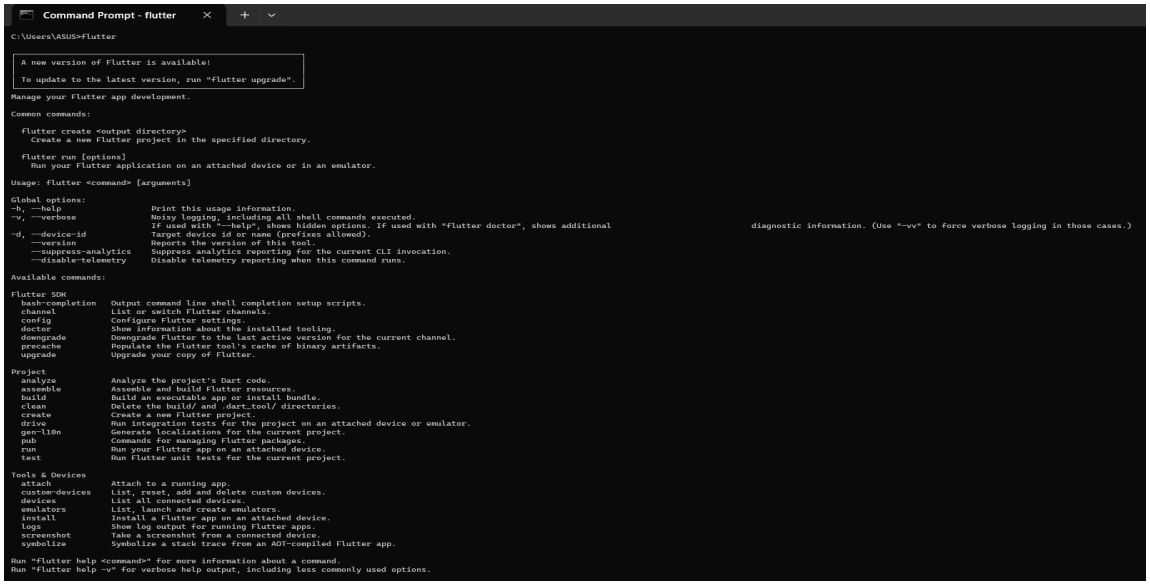
Step 2: Once the download is complete extract the zip file and place it in the desired folder.



Step 3: To run the Flutter command in the regular windows console, you need to update the system path to include the flutter bin directory. The following Steps are required to do this: Go to THIS PC -> Properties -> Advanced system settings -> Environment variables.

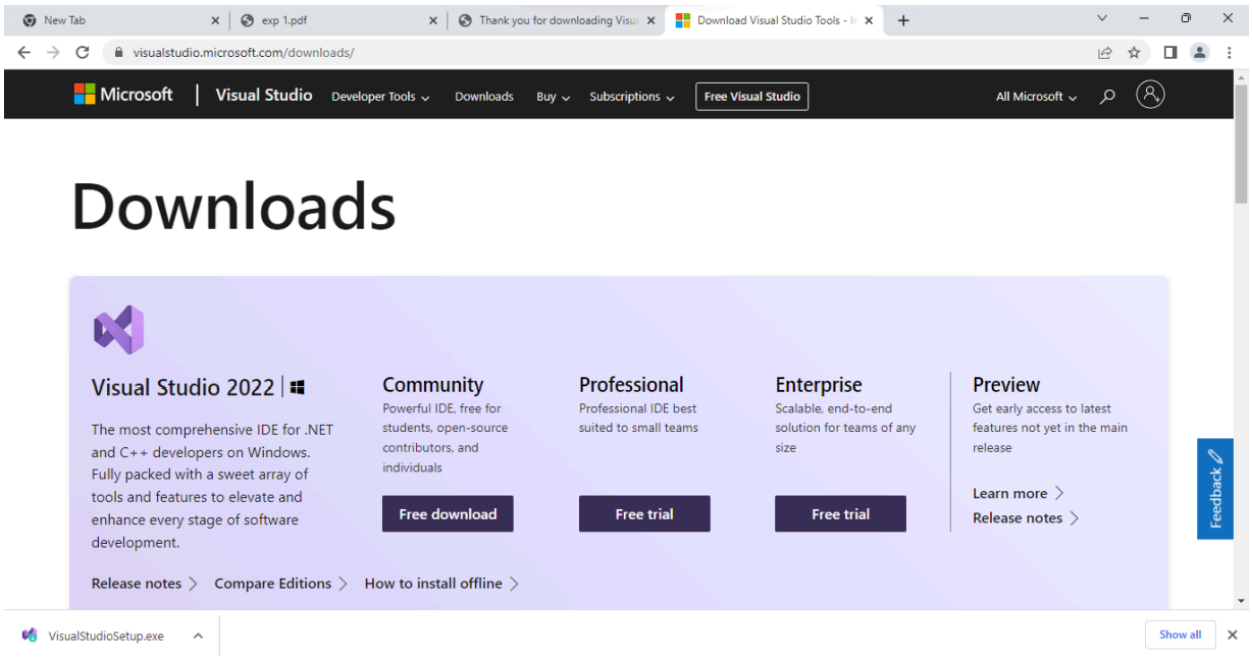
Step 4: Select the Path -> click on Edit and add the path to the Flutter bin folder.

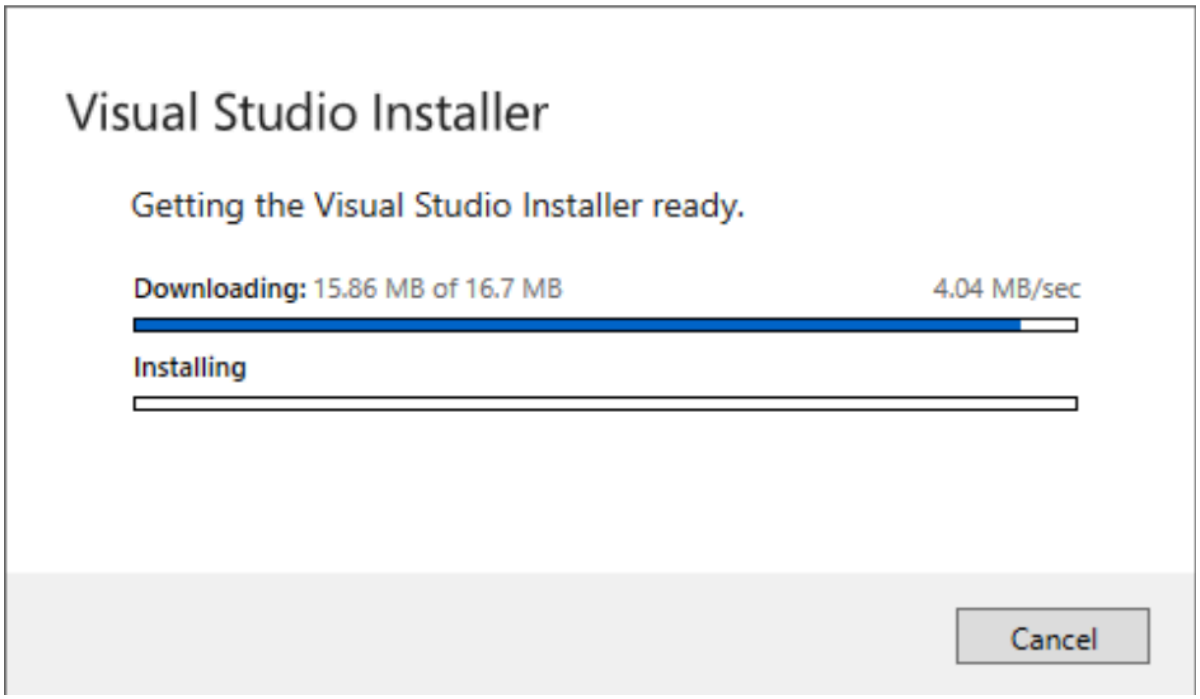
Step 5: Run the flutter command in the Command Prompt.



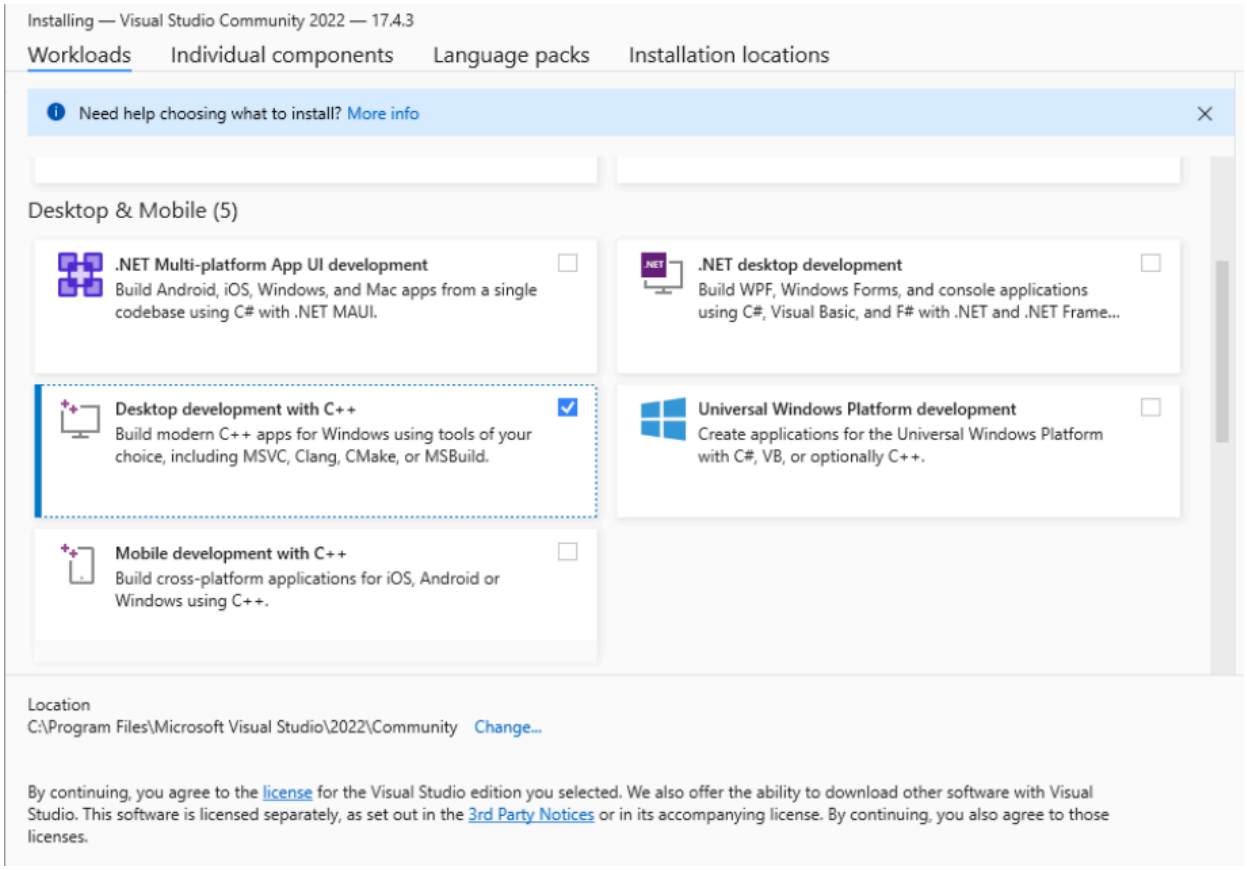
Also, run the flutter doctor command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

Step 6: Now we will resolve all the issues and install and add the missing tools required for Flutter app development.
Step 6.1: Download and install Visual Studio.



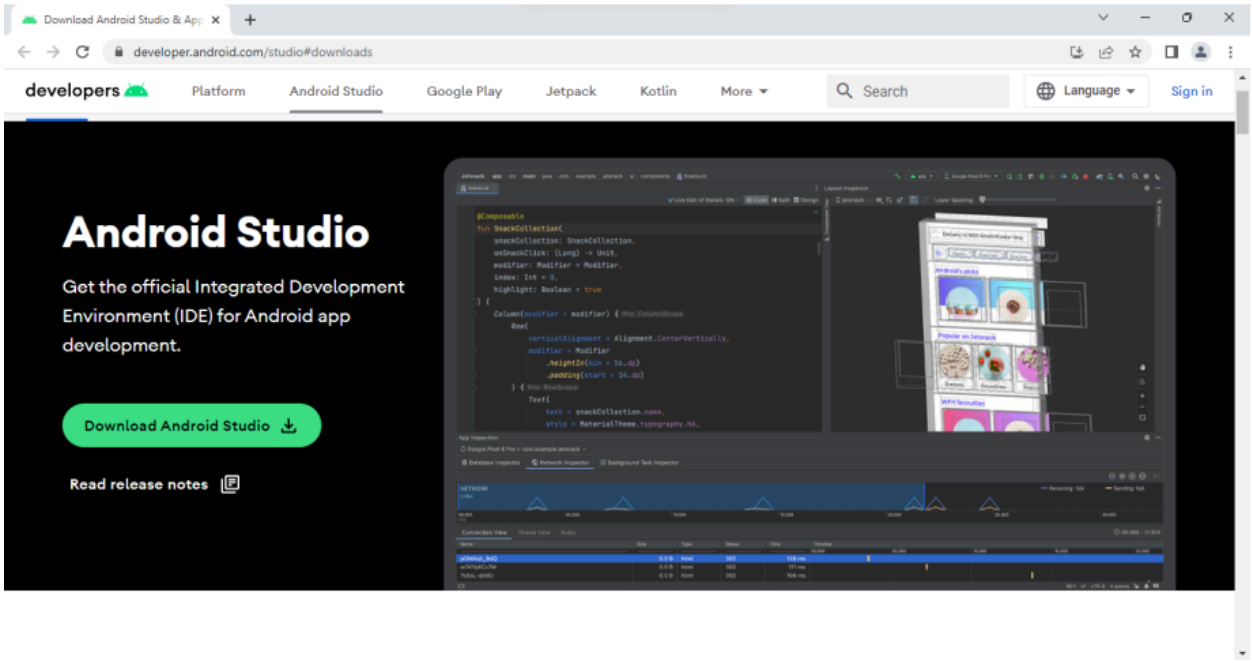


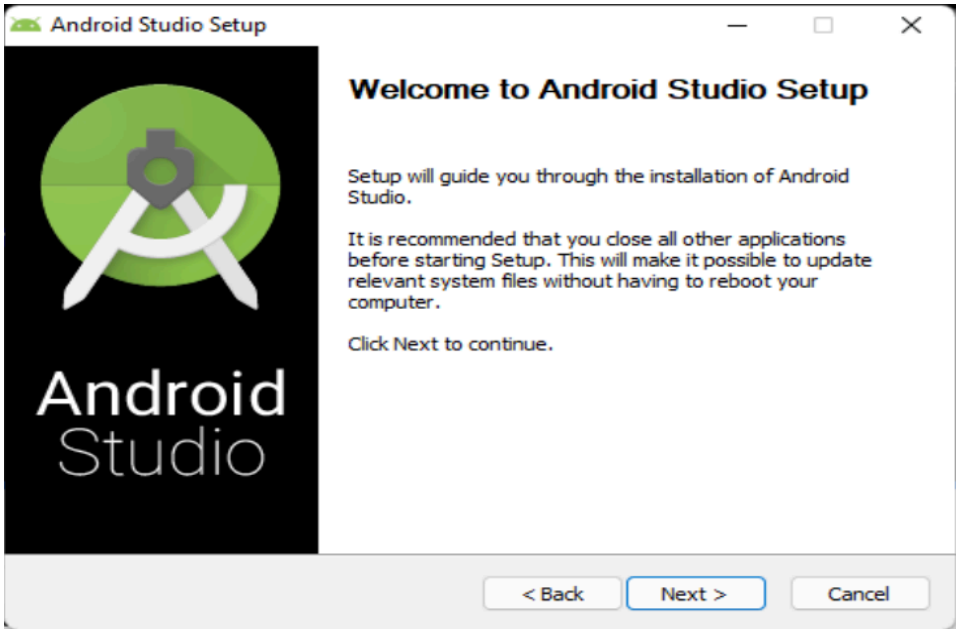
While Installing Visual Studio, make sure to select Desktop development with C++.

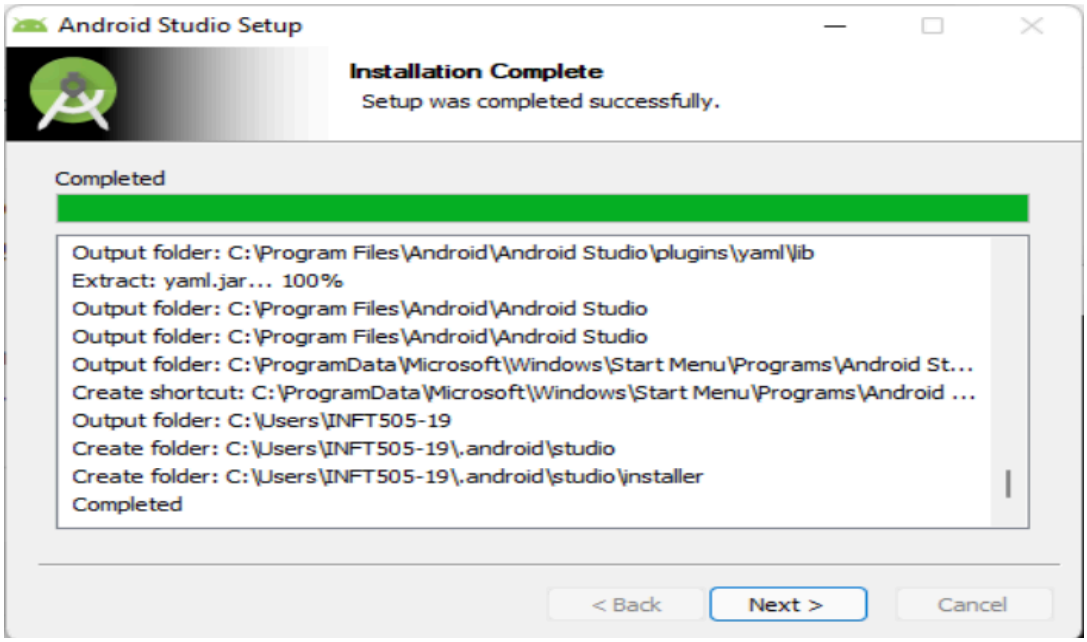


We can see that the Visual Studio issue has been solved.

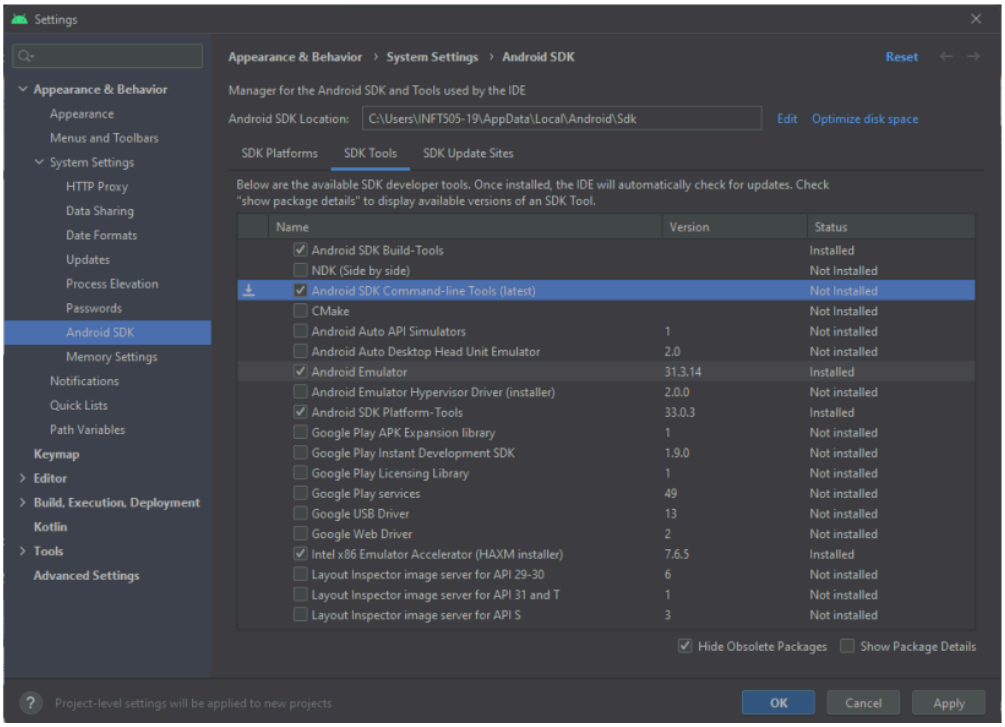
Step 6.2: Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE.


Download Android Studio & App
developer.android.com/studio#downloads
developers
Platform
Android Studio
Google Play
Jetpack
Kotlin
More
Search
Language
Sign in
Android Studio
Get the official Integrated Development Environment (IDE) for Android app development.
Download Android Studio
Read release notes


Android Studio Setup
Welcome to Android Studio Setup
Setup will guide you through the installation of Android Studio.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Android Studio
< Back
Next >
Cancel

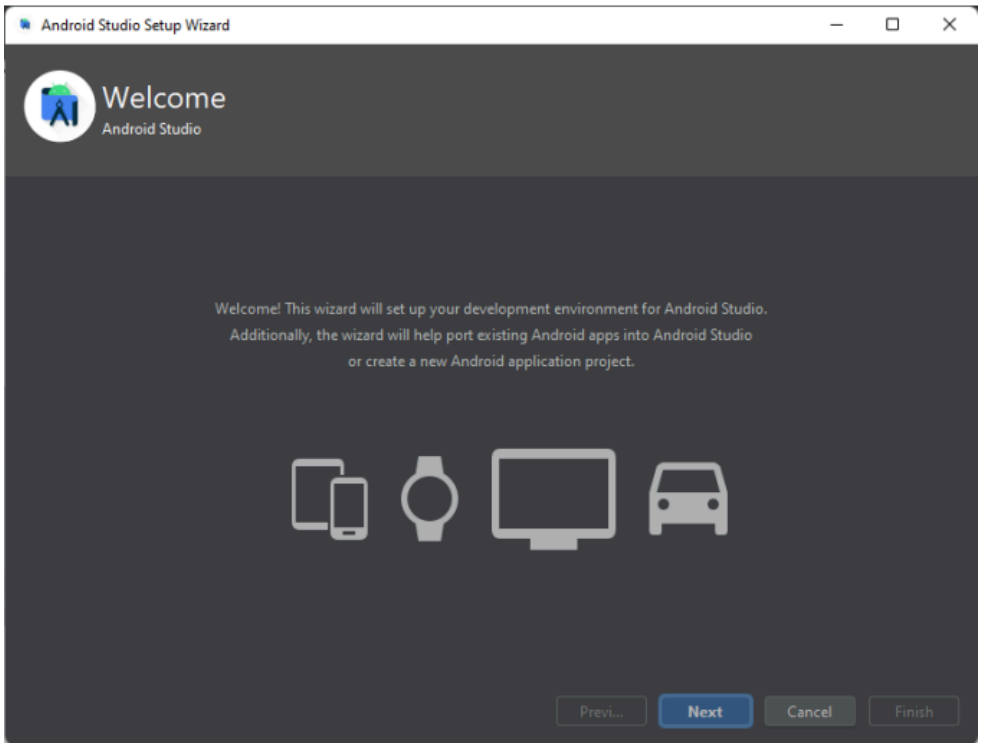


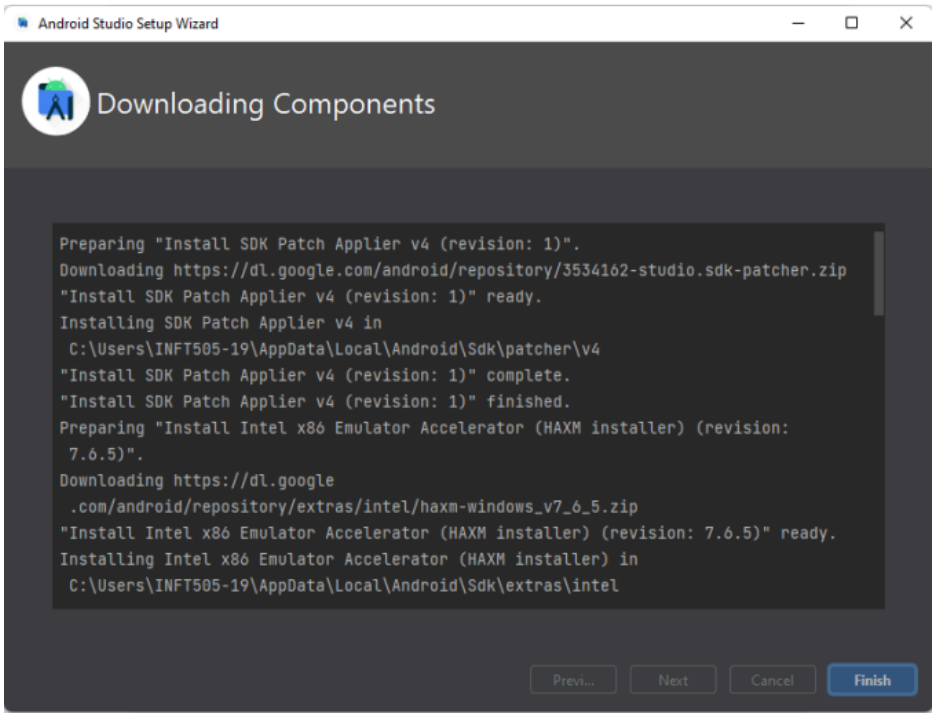
Once the android Studio is installed, Download the Android SDK Command-Line Tools present in the Android SDK section in Android Studio.



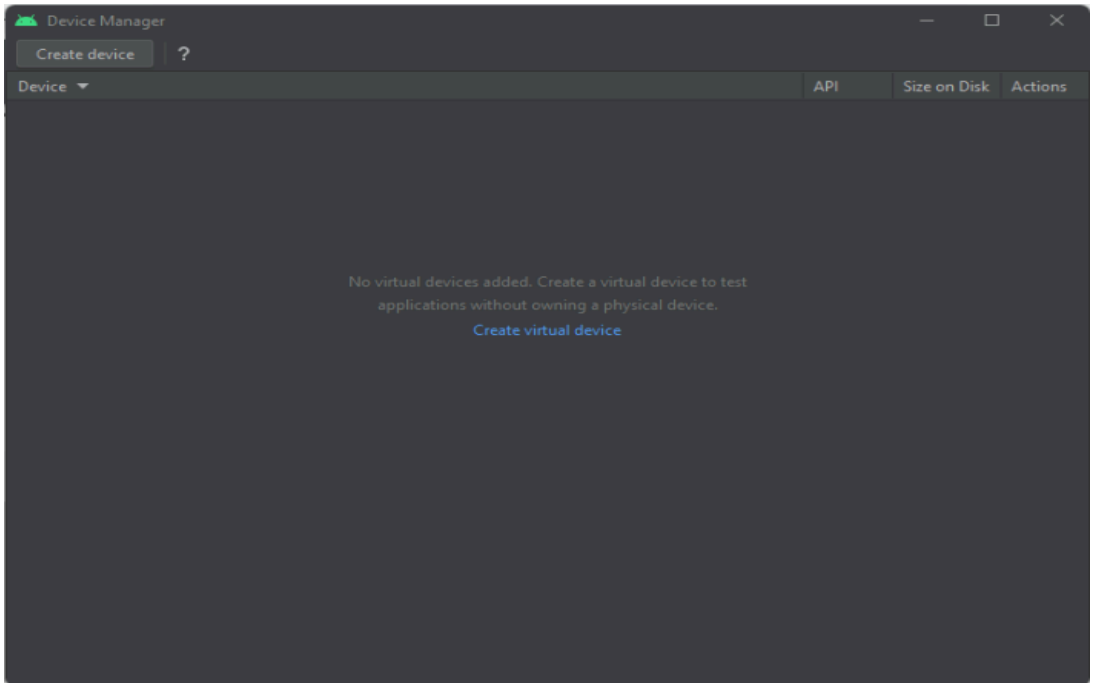
Run the flutter doctor command and run flutter doctor –android licenses to accept all the required licenses.

Step 7: Next, you need to set up an Android emulator. It is responsible for running and testing the Flutter application.

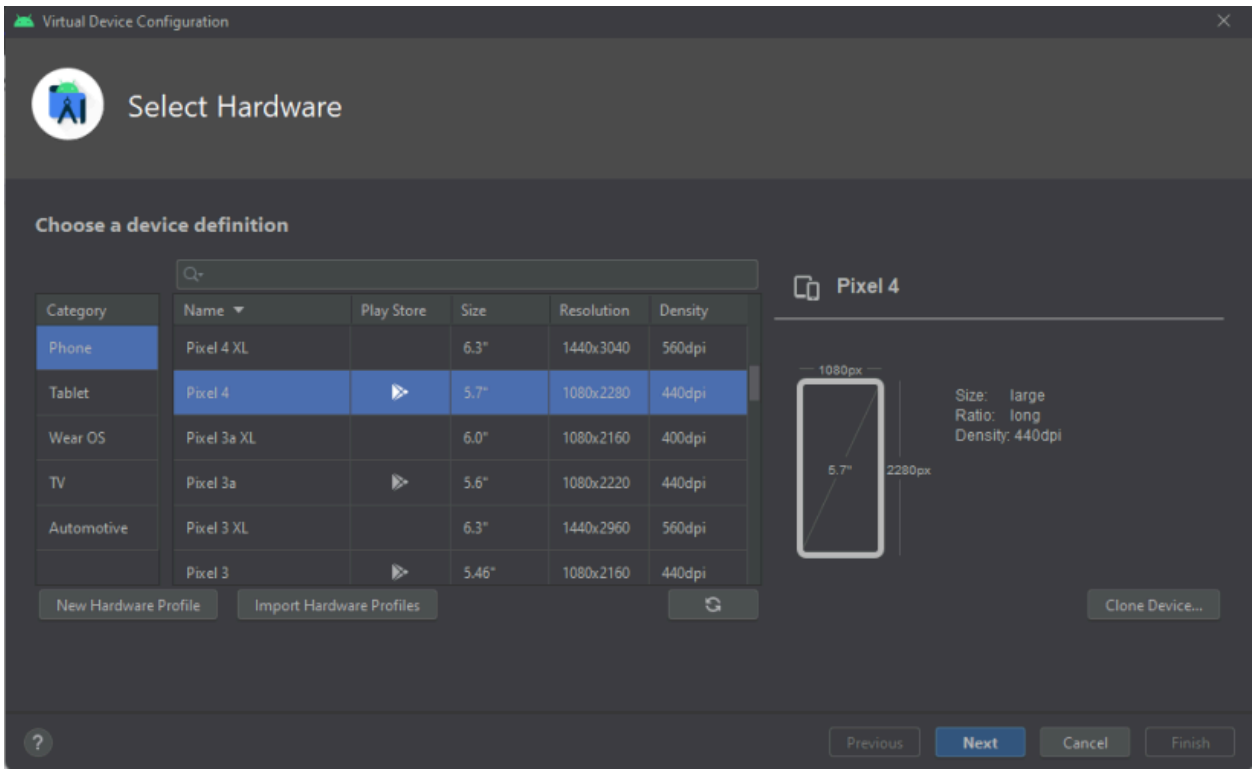




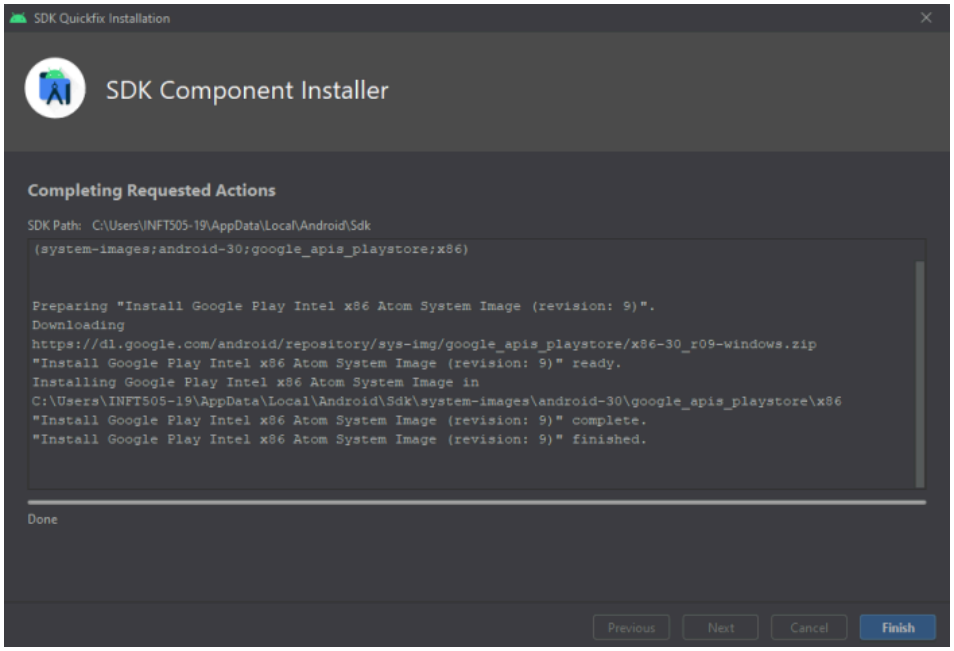
Step 7.1: To set an Android emulator, go to Android Studio > Tools > SDK Manager and select Create Virtual Device. You will get the following screen:



Step 7.2: Choose your device definition and click on Next.



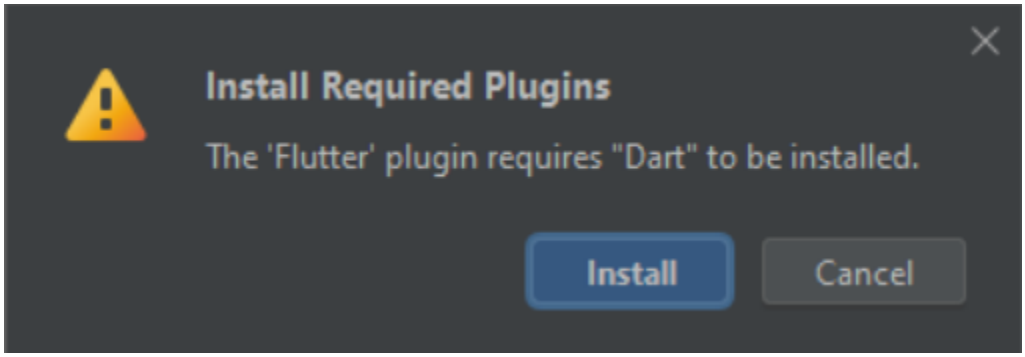
Step 7.3: Select and download the latest operating system for our Emulator and click on Finish.



Step 7.4: Click on the Run button and the following screen will be displayed.

Step 8: Now, install Flutter and Dart plugins for building the Flutter application in Android Studio. These plugins provide a template to create a Flutter application and give the option to run and debug the Flutter application in the Android Studio itself.

Open the Android Studio and then go to File->Settings->Plugins. Now, search the Flutter plugin. If found, select the Flutter plugin and click install. When you click on install, it will ask you to install Dart plugin as below screen. Click Install to proceed.



Finally when all these Steps are followed restart the Android Studio once and then your Flutter environment is successfully configured.

Conclusion:

In conclusion, the installation and configuration of Flutter and Android Studio proved to be a pivotal step in creating a robust mobile development environment. By seamlessly integrating these tools, developers gain a versatile platform for building cross-platform applications with efficiency. The synergy between Flutter's UI toolkit and Android Studio's comprehensive development features opens up a world of possibilities, empowering developers to create visually appealing and high-performance apps for both Android and iOS. As technology continues to evolve, mastering the setup of these tools becomes increasingly essential for staying at the forefront of mobile app development.