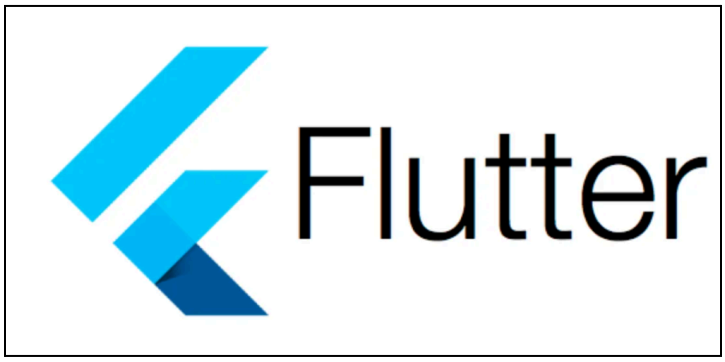
Experiment No.1
MAD & PWA LAB

Aim: Installation and Configuration of Flutter Environment.

Theory:

Introduction:

Flutter is Google's Mobile SDK to build native iOS and Android apps from a single codebase. It was developed on December 4, 2018. When building applications with Flutter, everything is towards Widgets – the blocks with which the Flutter apps are built. The User Interface of the app comprises many simple widgets, each handling one particular job.



Advantages:

1.Single Codebase for Multiple Platforms: Flutter allows you to write code once and run it on multiple platforms, including iOS, Android, web, and desktop. This can significantly reduce development time and effort compared to maintaining separate codebases for each platform.

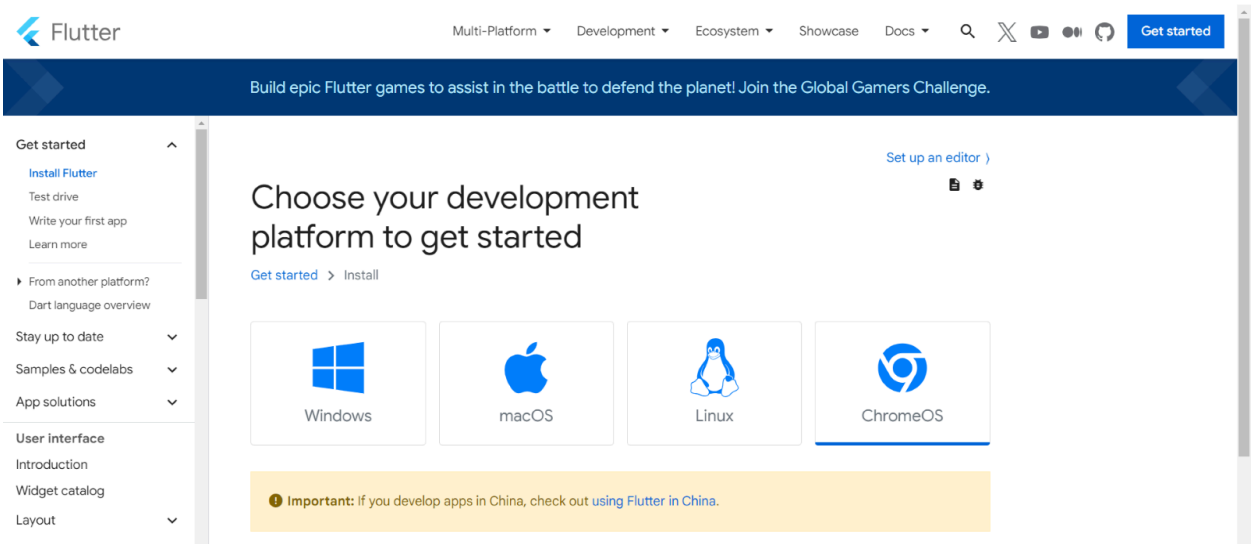
2.Hot Reload: One of Flutter's standout features is its hot reload capability. Developers can see the impact of their code changes almost instantly, making the development process more iterative and efficient.

3.Expressive and Flexible UI: Flutter provides a rich set of customizable widgets that allow developers to create expressive and flexible user interfaces. The widget-based architecture enables the creation of complex UIs with ease.

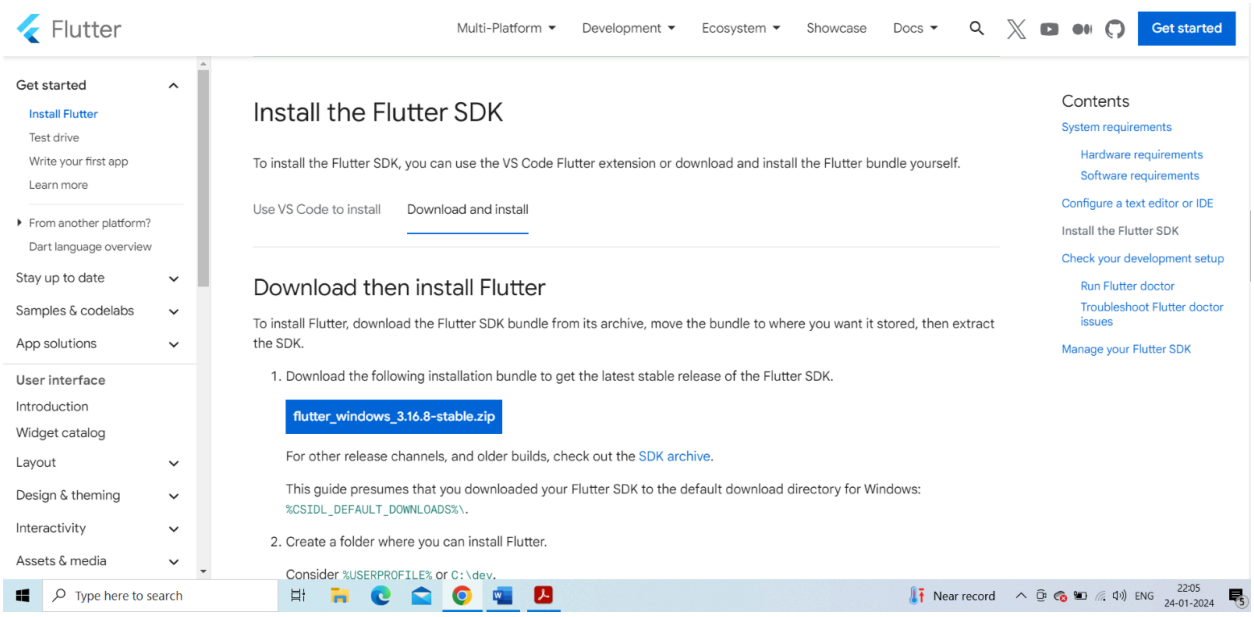
4.Performance: Flutter compiles to native ARM code and doesn't rely on an intermediate interpretation layer, leading to high performance. This is crucial for delivering smooth and responsive user experiences.

1. Install the Flutter SDK:

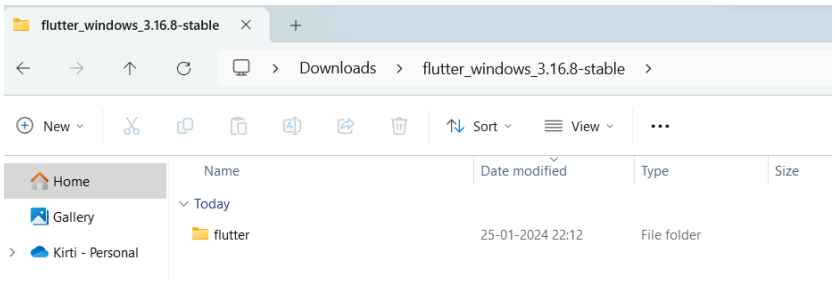
Step 1: Visit the official Flutter Software Development Kit website at https://docs.flutter.dev/get-started/install to access the installation bundle for Windows. Once on the website, you'll encounter the following screen.



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

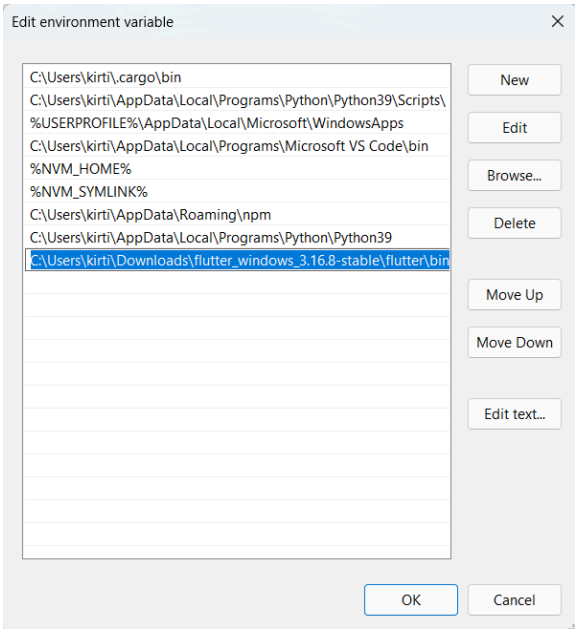


Step 3: After downloading, unzip the file and place it in your preferred installation folder or location, such as C:/Flutter.



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.

```
C:\Users\kirti\Downloads>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]
```

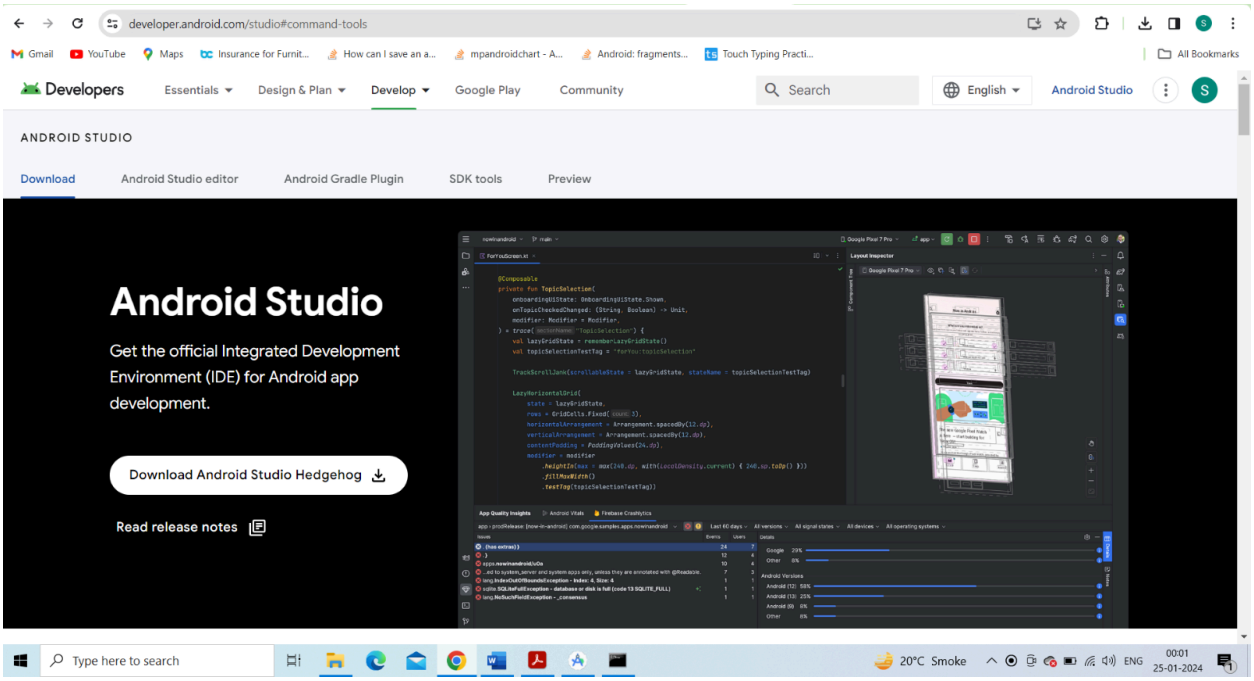
Execute the "flutter doctor" command to assess the status of your Flutter installation by checking for all the necessary requirements for Flutter app development. This command generates a comprehensive report outlining the current state of your Flutter setup.

```
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!
```

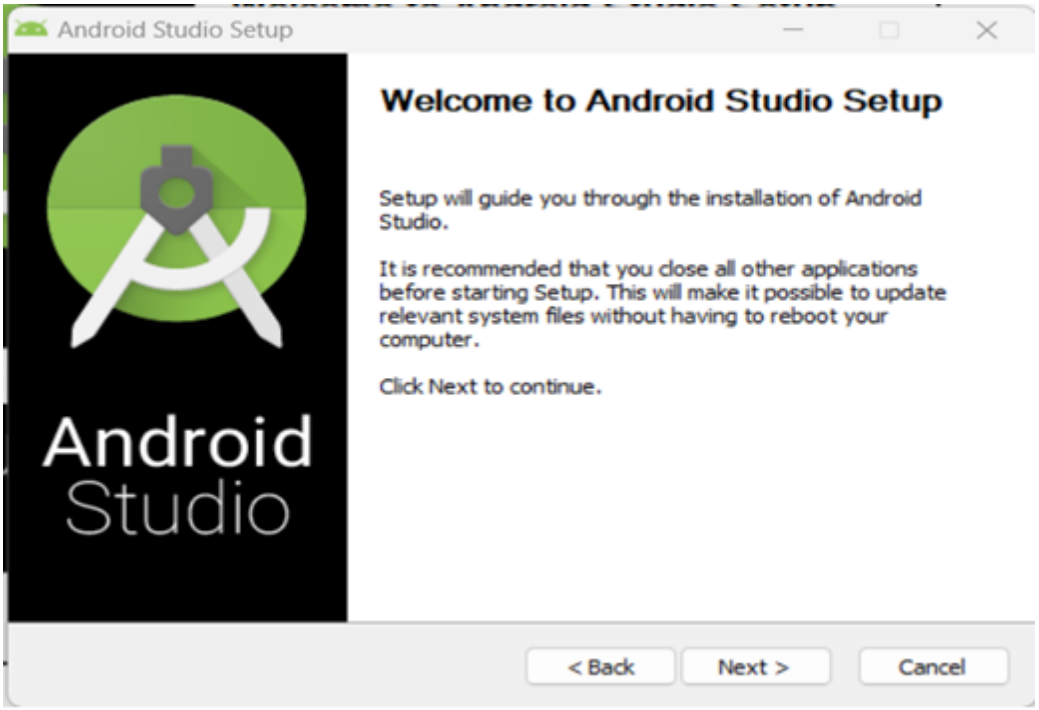
Step 6: Upon executing the given command, a system analysis will be conducted, presenting a comprehensive report. This report will include information on any absent tools necessary for Flutter operation, along with details on available development tools that may not be linked to the device, as illustrated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

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.]

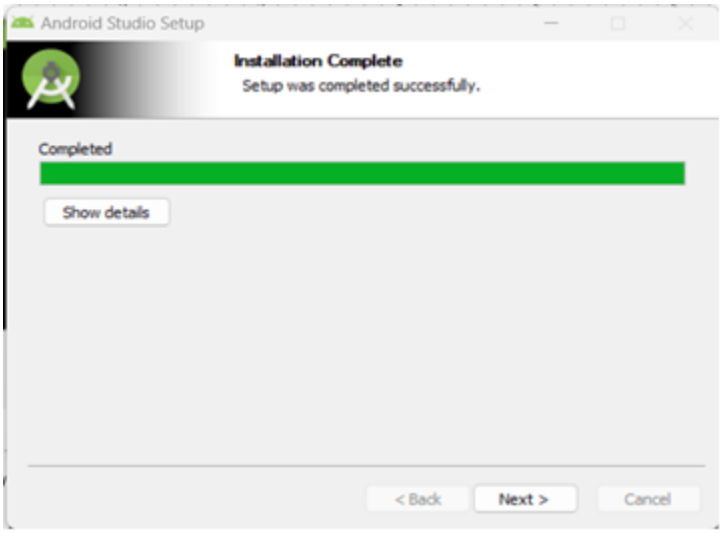
Step 7.1: Retrieve the most recent Android Studio executable or zip file directly from the official website.



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.

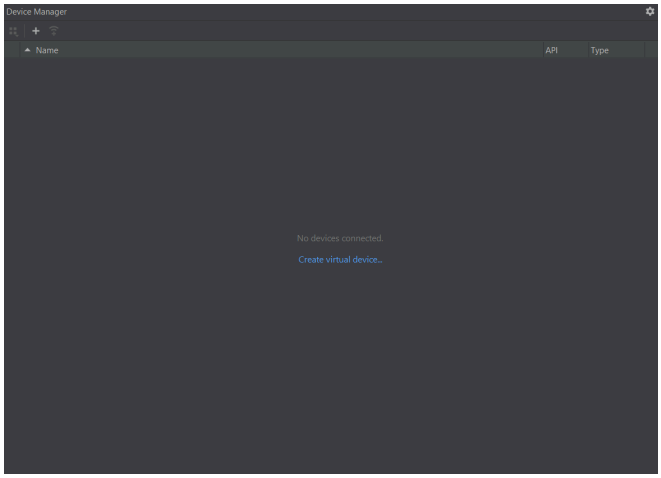


Step 7.4: Execute the command $ flutter doctor and then proceed to run $ flutter doctor --android-licenses.

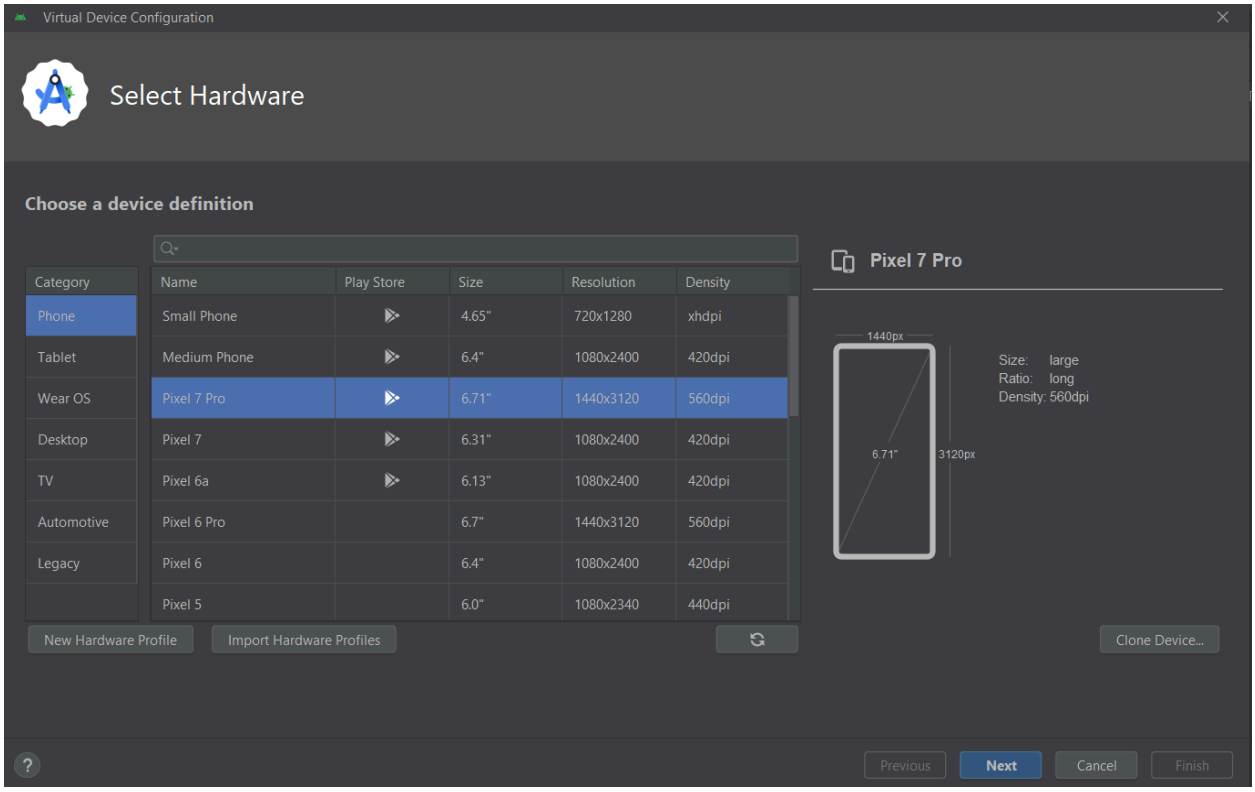
```
C:\Users\HP>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.16.8, on Microsoft Windows [Version 10.0.19045.3930], locale en-IN)
[√] Windows Version (Installed version of Windows is version 10 or higher)
[√] Android toolchain - develop for 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!
```

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.

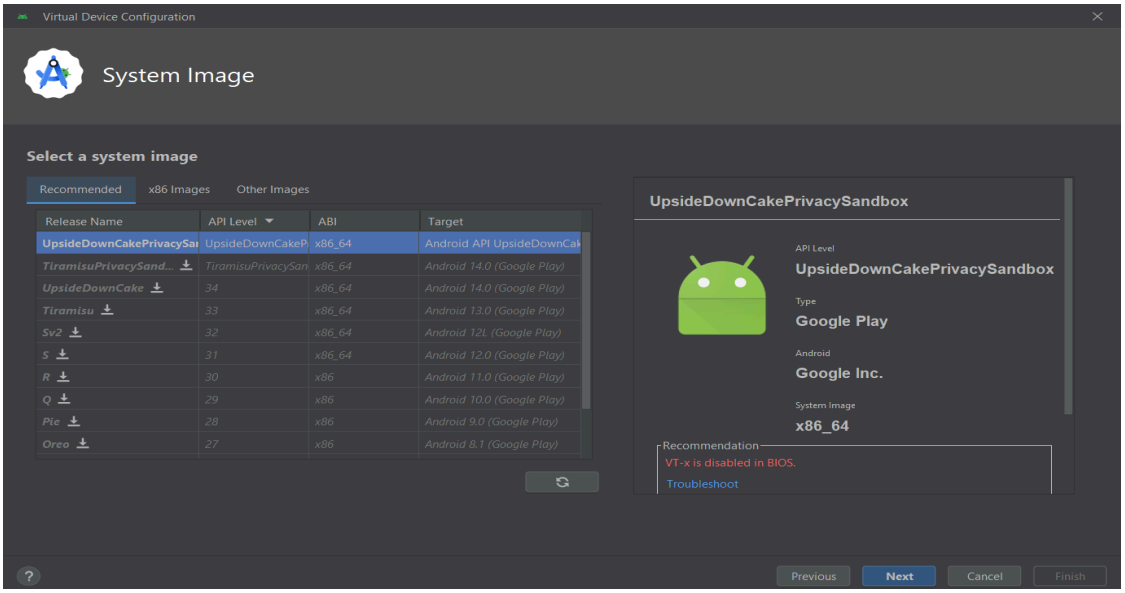


Step 8.2: Select the device definition and proceed by clicking on Next.



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

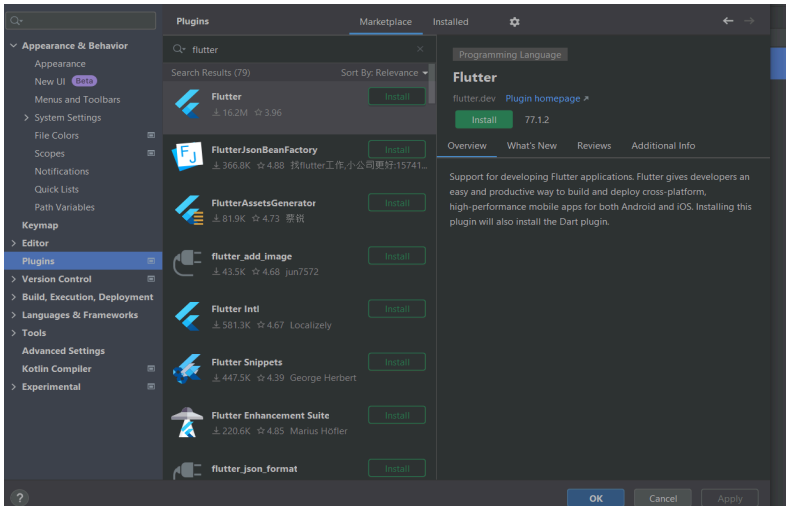
Step 8.4: If the AVD configuration is accurate, proceed to click on Finish, and the subsequent screen will be displayed.



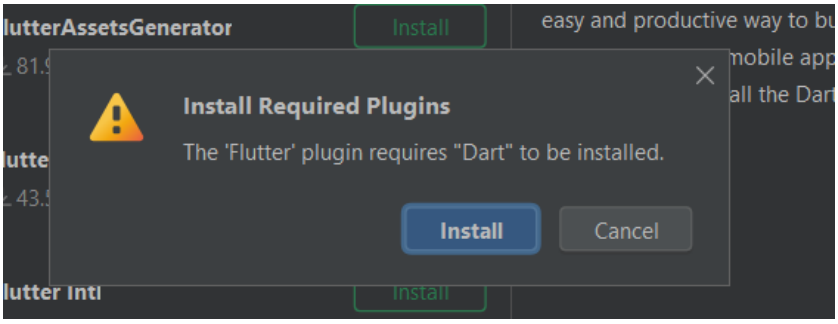
Step 8.5: After running the created virtual device, The Android emulator is displayed as below screen.

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.

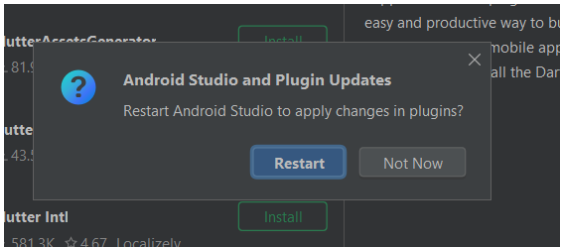
Step 9.1: Launch Android Studio, navigate to Settings by selecting File, then go to Plugins.



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 learned to set up the Flutter environment, install the Flutter SDK, configure Android Studio, and create a virtual device for Android development, ensuring proper SDK tool installation and license acceptance.