Experiment No.1

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

- **Theory:**

Flutter is an open-source UI software development toolkit created by Google. It is used to build natively compiled applications for mobile, web, and desktop from a single codebase. Flutter allows developers to use a single codebase to create high-quality, high-performance applications for multiple platforms, eliminating the need to write separate code for iOS and Android.

Key features of Flutter include:

- **Dart Programming Language:** Flutter apps are written in Dart, a modern, object-oriented programming language also developed by Google.
- **Widget-based Framework:** Flutter uses a widget-based framework where everything in the UI is a widget. Widgets are combined to create complex UIs, and the framework provides a rich set of pre-designed widgets for common interface elements.
- **Hot Reload:** One of the standout features of Flutter is its Hot Reload capability, which allows developers to see the results of code changes instantly, without restarting the application. This greatly speeds up the development process and enhances productivity.
- **Expressive UI:** Flutter provides a rich set of customizable widgets that allow developers to create expressive and visually appealing user interfaces.
- **Single Codebase:** Developers can write code once and run it on multiple platforms, including iOS, Android, web, and desktop. This helps in reducing development time and maintenance efforts.
- **Performance:** Flutter apps are compiled to native machine code, resulting in high performance and smooth animations.
- **Community and Ecosystem:** Flutter has a growing and active community, and it has gained popularity for mobile app development. The ecosystem includes various plugins and packages that developers can use to extend the functionality of their applications.
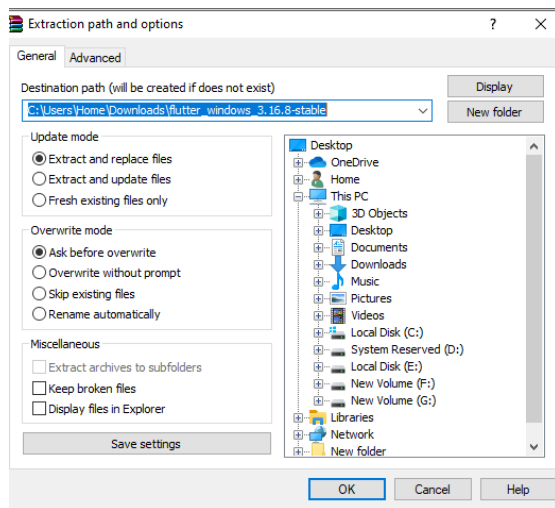
- **Install the Flutter SDK:**

  Step: 1   Visit the official Flutter Software Development Kit website at
  https://docs.flutter.dev/get-started/install and click on windows.
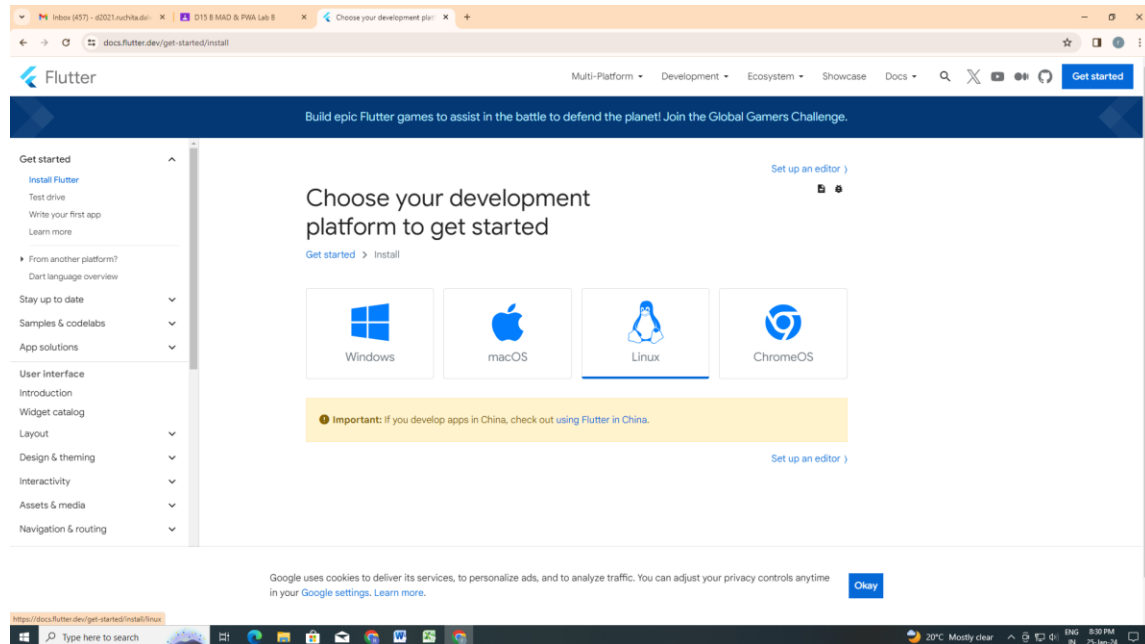
  Step: 2   Click on mobile , where you'll discover the SDK download link.
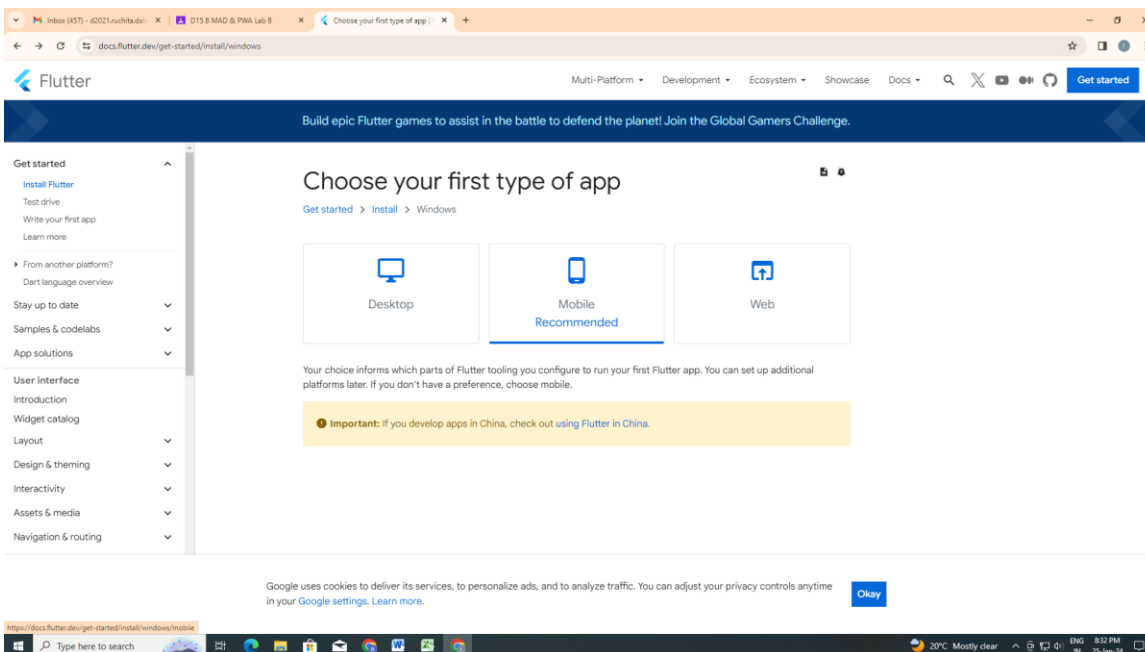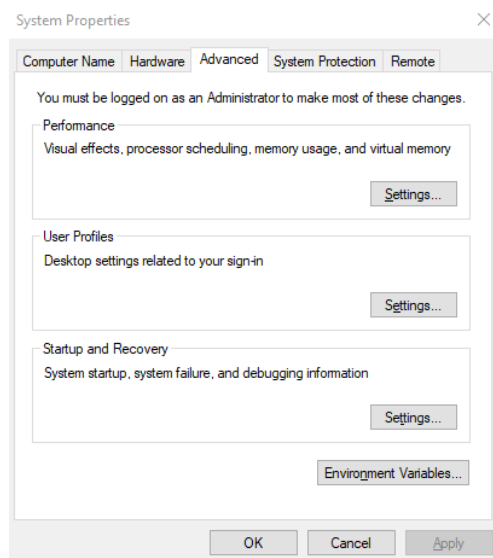

Step: 3   After downloading, unzip the file.

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: Go to system properties.

Step 4.2: Go to environment variables. Choose the "Edit" option after selecting the path, and you will be directed to the ensuing screen.
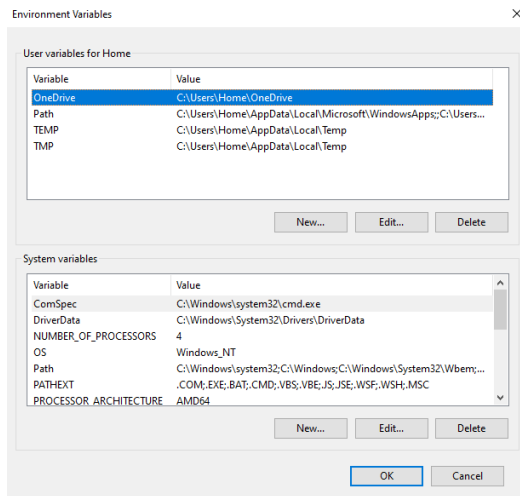
Step 4.3: Select New option in the current window, then input the path of the Flutter bin folder in the Variable Value field. Click on OK.
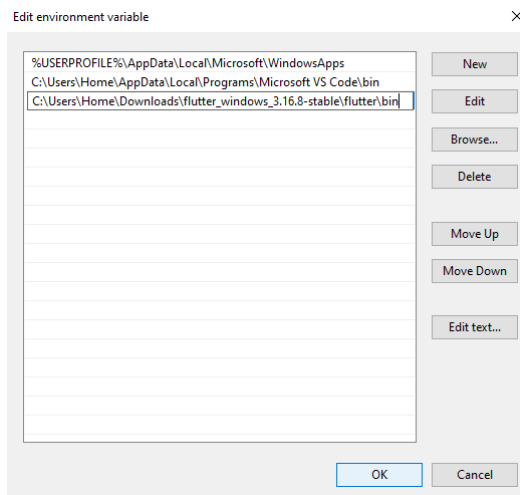
Step 5: Execute the command $ flutter in the command prompt.

Step 6: 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.


When you execute the provided command, it will conduct a system analysis and generate a report, displaying information about missing tools essential for running Flutter. The report will also outline available development tools that are not currently linked with the device. This information is presented in a detailed format, as illustrated in the accompanying image.

Step 7: To set up the Android SDK, install the Android Studio IDE if the flutter doctor command doesn't detect the Android SDK tool on your system. Follow these steps to install the Android Studio IDE.

Step 7.1: Download the latest Android Studio executable or zip file from the official site (https://developer.android.com/studio).
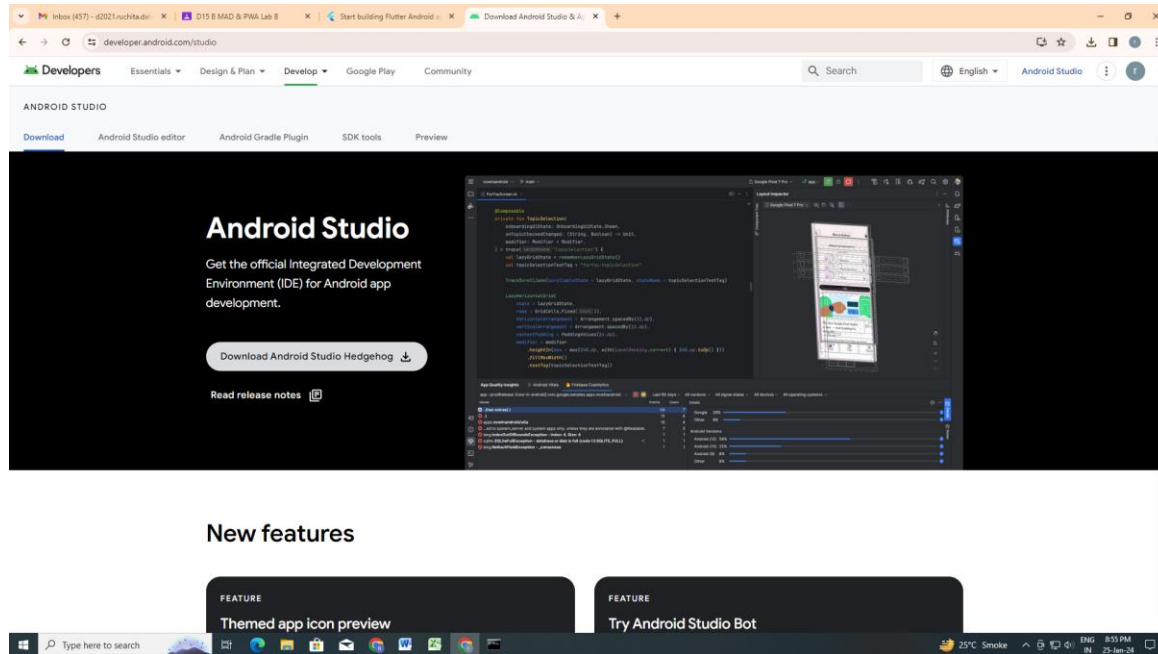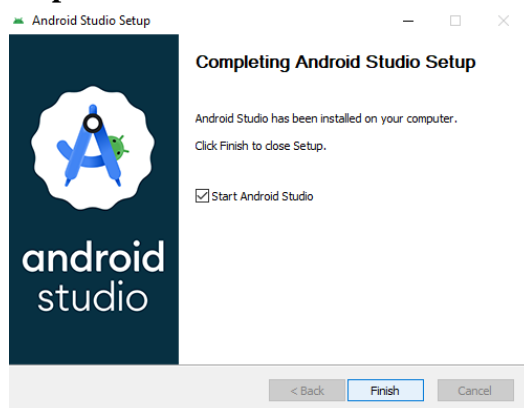

Step 7.2: Upon completion of the download, proceed to open and run the .exe file. A dialog box will then appear go on following the steps of the installation wizard.
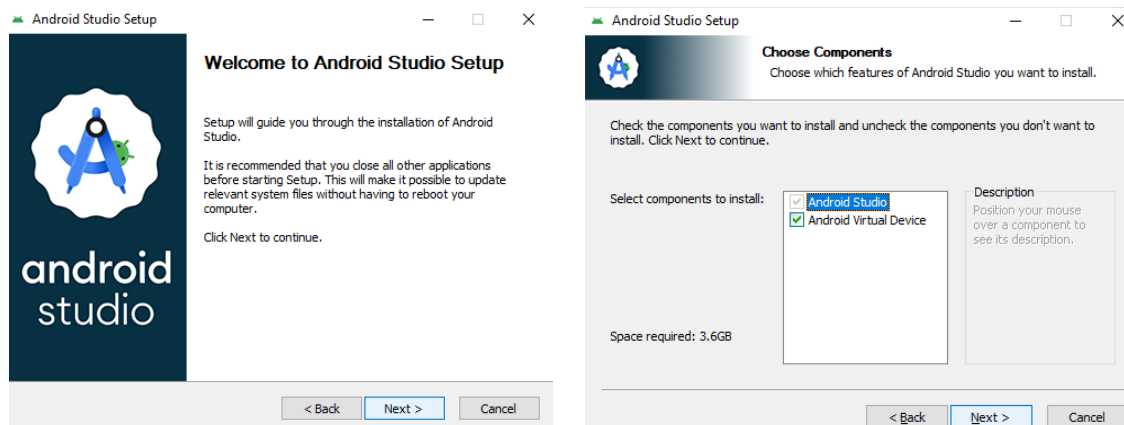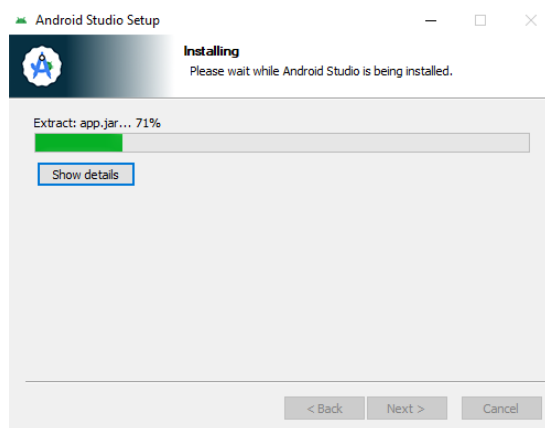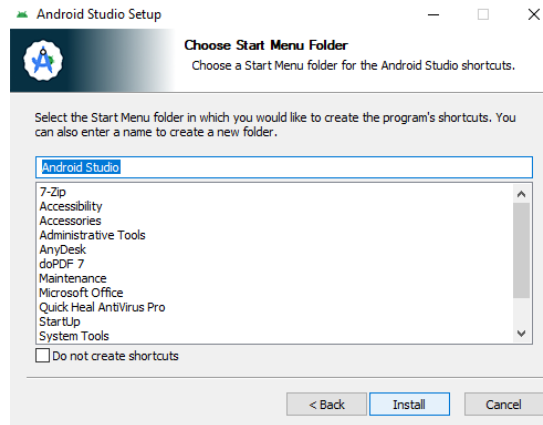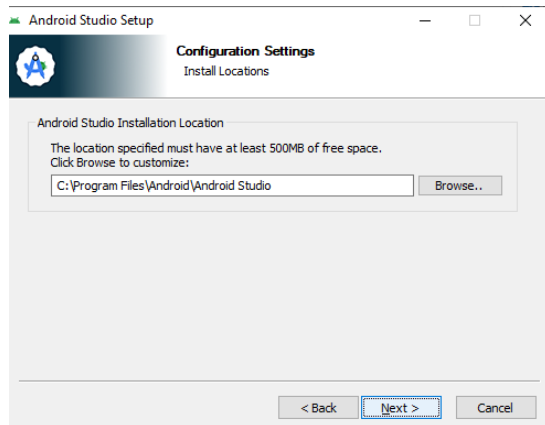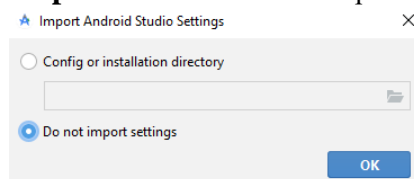
Step 7.3: Once the installation wizard completes, you will get the following screen click on finish.

Step 7.4: Choose '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.
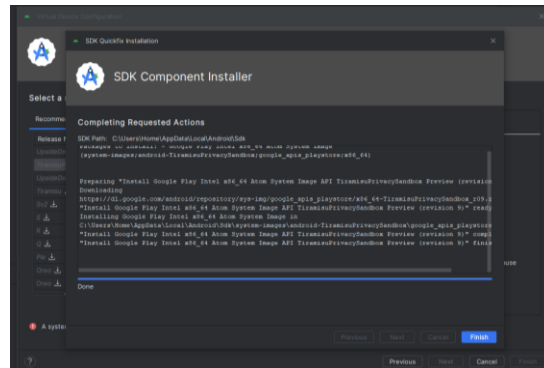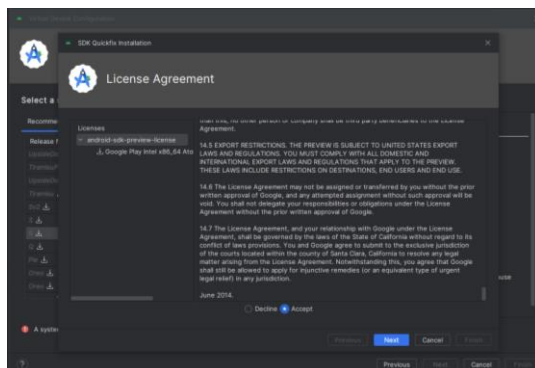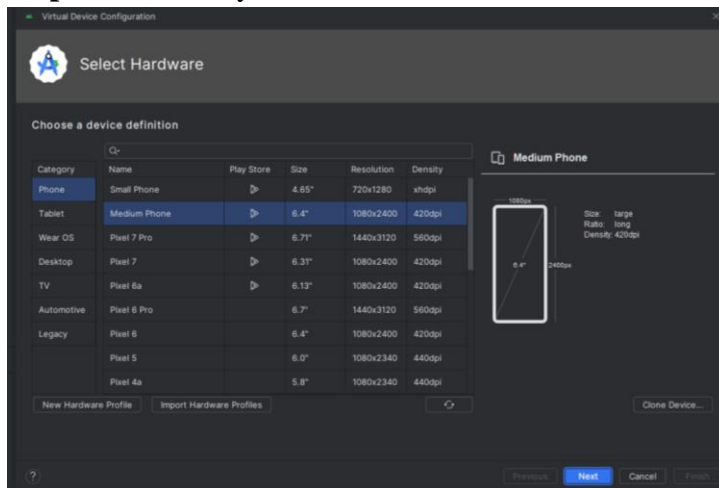

**Step 8:** Set up an Android emulator..

Step 8.1: Generate a virtual Android emulator by going to Android Studio, then choosing Tools > Android > AVD Manager. Alternatively, you can access it by navigating to Help -> Find Action, and entering "Emulator" in the search box to access the relevant screen.
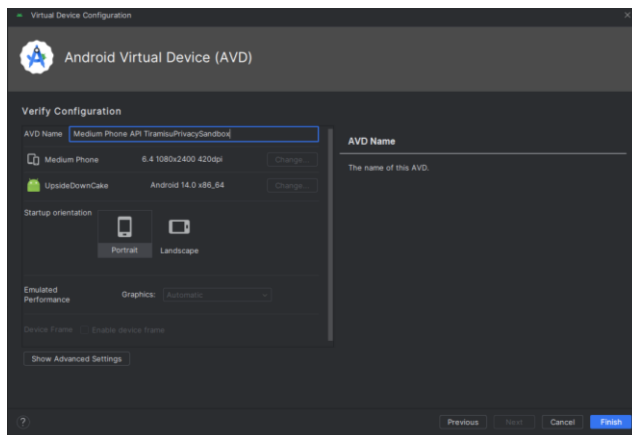
Step 8.2: Choose your device definition and click on Next and accept the agreement.
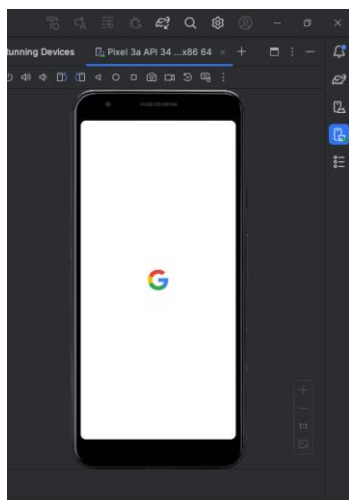
Step 8.3: Select the system image for the latest Android version and click on Next.
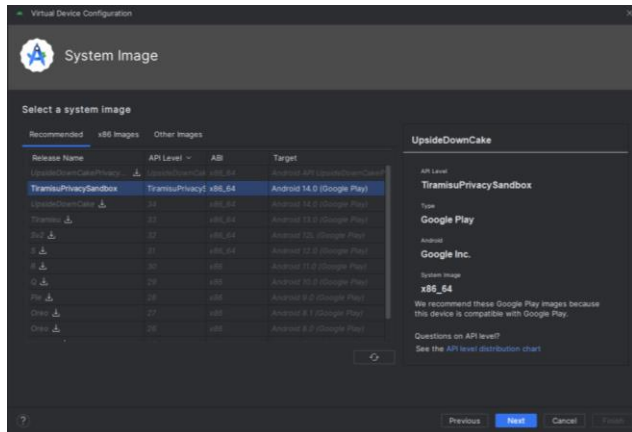
**Step 8.4:** Now, verify the all AVD configuration. If it is correct, click on Finish.

Step 8.5: After running the created virtual device, The Android emulator is displayed as below screen.
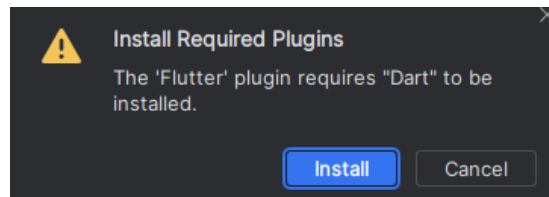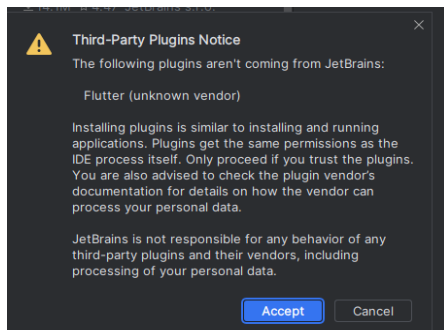
Step 9: configure the Flutter and Dart plugins for Android Studio to facilitate the development of Flutter applications. Follow these steps:
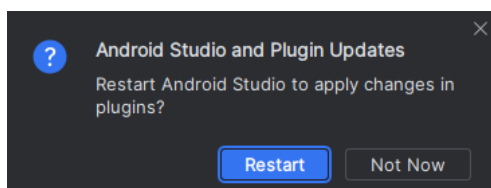
Step 9.1: In Android Studio and then go to File and then click on Settings there you will find Plugins.
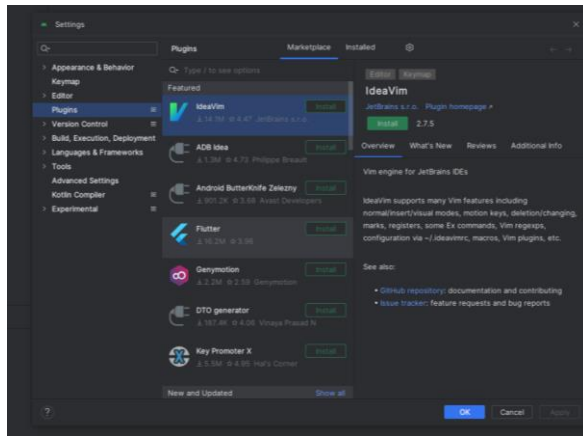
Step 9.2: Accept agreement and click on install.

Step 9.2: Restart IDE.

- **<u>Conclusion:</u>**

Therefore, we now have a grasp of the steps involved in installing and configuring the Flutter environment. This includes installing the Flutter SDK, configuring Android Studio, and completing the setup by creating and integrating a virtual device within Android Studio.