

Name: Param Keswani
Division: D15A
Roll No:27
Batch: B

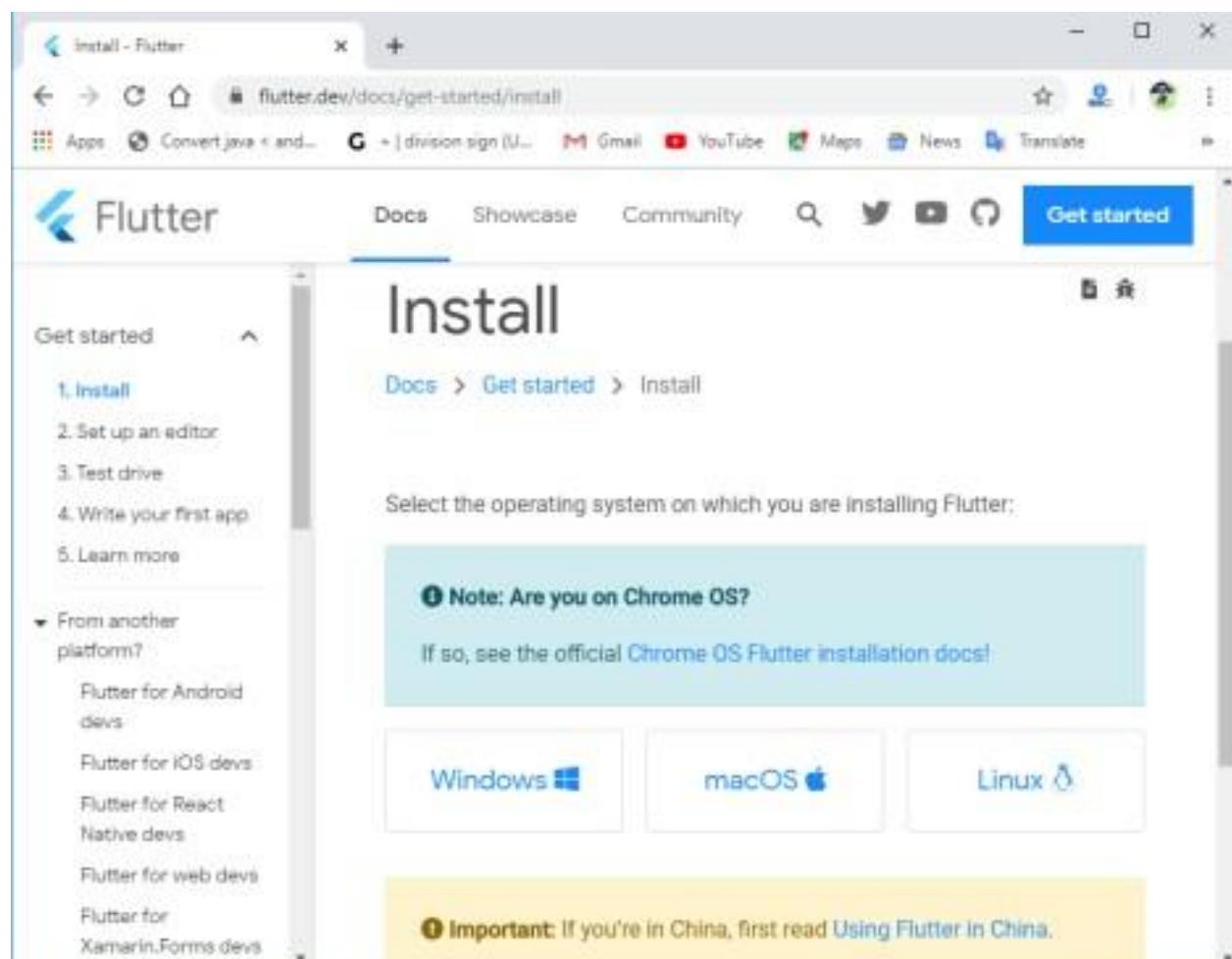
Experiment No. 1

Aim: To Install and Configure Flutter Environment

Pre Requisites:

Install the Flutter SDK

Step 1: Download the installation bundle of the Flutter Software Development Kit for windows. To download Flutter SDK, Go to its official [website https://docs.flutter.dev/get-started/install](https://docs.flutter.dev/get-started/install), you will get the following screen.

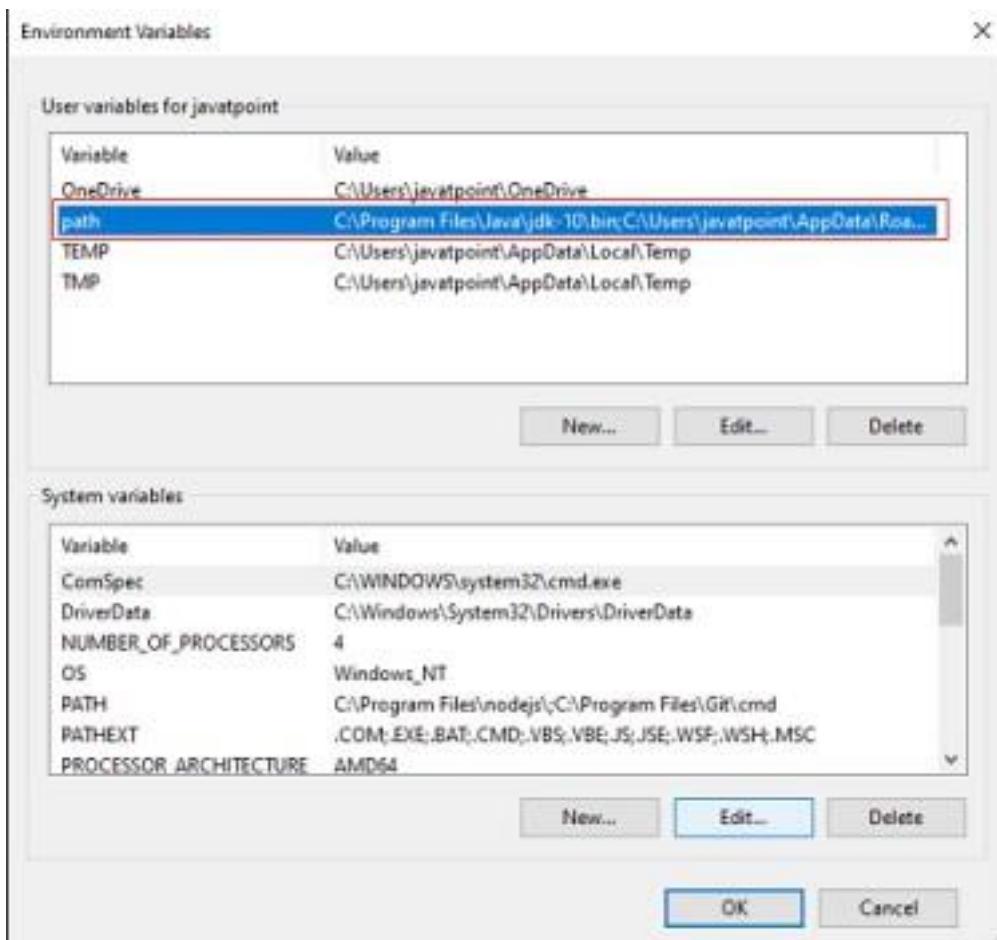


Step 2: Next, to download the latest Flutter SDK, click on the Windows icon. Here, you will find the download link for [SDK](#).

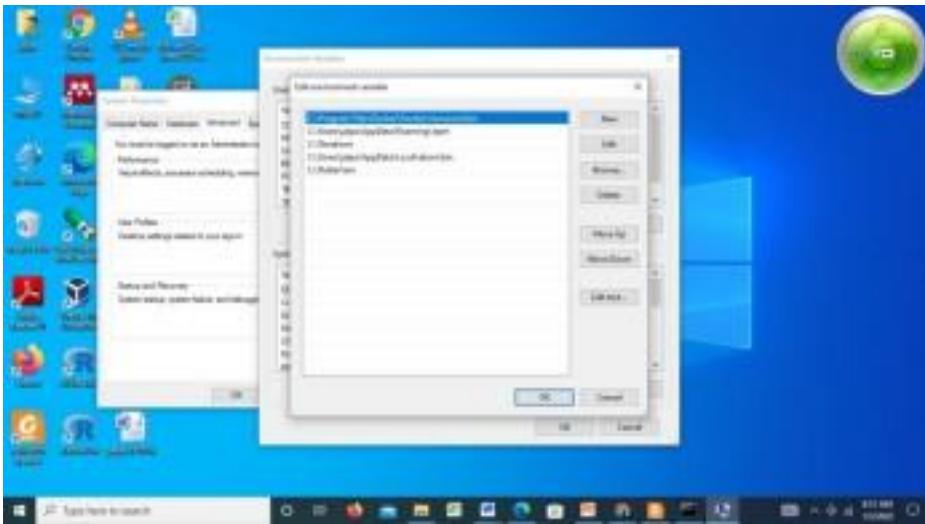
Step 3: When your download is complete, extract the **zip** file and place it in the desired installation folder or location, for example, C:/Flutter.

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.

```
Microsoft Windows [Version 10.0.19043.1435]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jalpa\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.
  --no-prefs           If used with '--help', show hidden options. If used with "flutter doctor", show additional
  info. (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.
  --suppress-analytics suppress analytics reporting when this command runs.

Available commands:
  flutter doc          Output command line shell completion setup scripts.
  channel              List or switch Flutter channels.
  config               Configure Flutter settings.
  doctor               View information about the installed tools.
  upgrade              Upgrade Flutter to the last active version for the current channel.
  precache              Populate the Flutter tool's cache of binary artifacts.
  upgrade              Upgrade your copy of Flutter.

Project:
  analyze              Analyze the project's Dart code.
  assemble             Assemble and build Flutter resources.
  build                Build an executable app or install bundle.
  clean                Delete the build/ and .dart_tool/ directories.
  create               Create a new Flutter project.
  test                 Run integration tests for the project on an attached device or emulator.
  format               Format one or more Dart files.

Type here to search
```

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.

```
See Google's privacy policy:  
https://policies.google.com/privacy  
  
C:\Users\jaisip> flutter doctor  
Doctor summary (to see all details, run flutter doctor -v):  
[!] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19043.1465], locale en-US)  
  • Android toolchain - develop for Android devices  
    ✘ Unable to locate Android NDK.  
      Install Android Studio from: https://developer.android.com/studio/index.html  
      On first launch it will assist you in installing the Android NDK components.  
      (or visit https://flutter.dev/docs/get-started/install/windows/android-setup for detailed instructions).  
  • If the Android NDK has been installed to a custom location, please use  
    'flutter config --android-ndk' to update to that location.  
  
  • Chrome - develop for the web  
  • Android Studio (not installed)  
  • VS Code (version 1.64.2)  
  • Connected device (2 available)  
  
Doctor found issues in 2 categories.  
  
C:\Users\jaisip> flutter doctor  
Doctor summary (to see all details, run flutter doctor -v):  
[!] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19042.1465], locale en-US)  
  • Android toolchain - develop for Android devices (Android NDK version 22.1.8431489)  
    ✘ cmdline-tools component is missing  
      Run "C:\Program Files\Android\Android Studio\ cmdline-tools\latest"  
      See https://developer.android.com/studio/command-line for more details.  
  • android license status unknown  
    Run "flutter doctor --android-licenses" to accept the SDK licenses.  
    See https://flutter.dev/docs/get-started/install/windows/android-setup for more details.  
  • Chrome - develop for the web  
  • Android Studio (version 2020.3)  
  • VS Code (version 1.64.2)  
  • Connected device (2 available)  
  
Doctor found issues in 3 categories.
```

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.

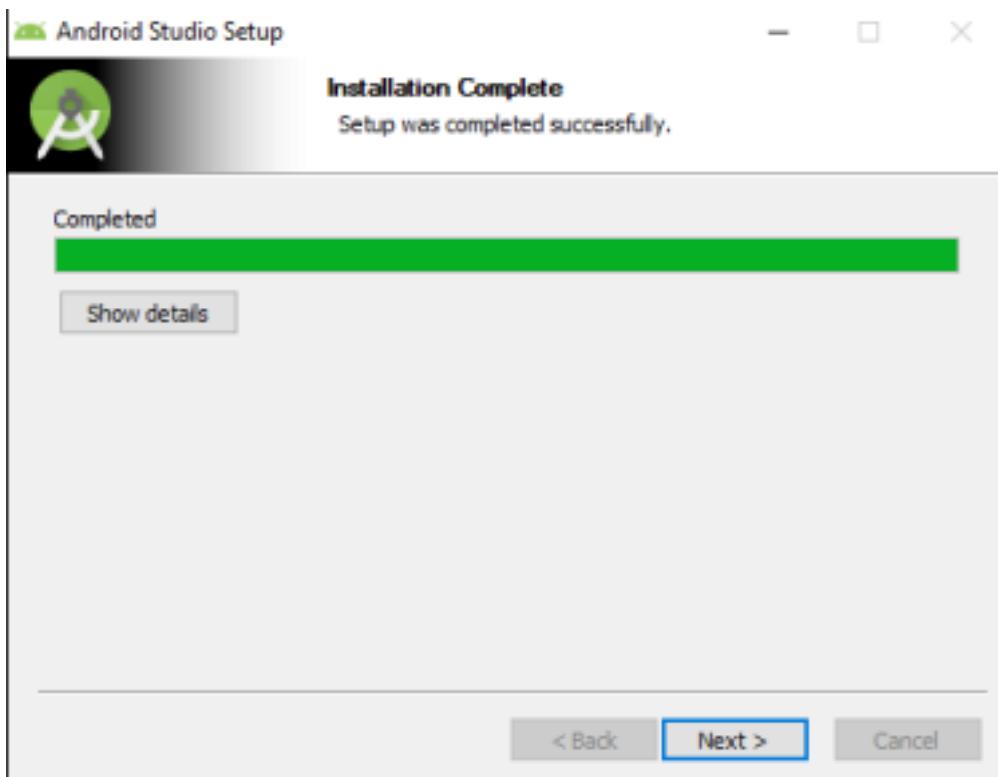
Step 7: Install the Android SDK. If the flutter doctor command does not find the Android SDK tool in your system, then you need first to install the Android Studio IDE. To install Android Studio IDE, do the following steps.

Step 7.1: Download the latest Android Studio executable or zip file from the [official site](#).

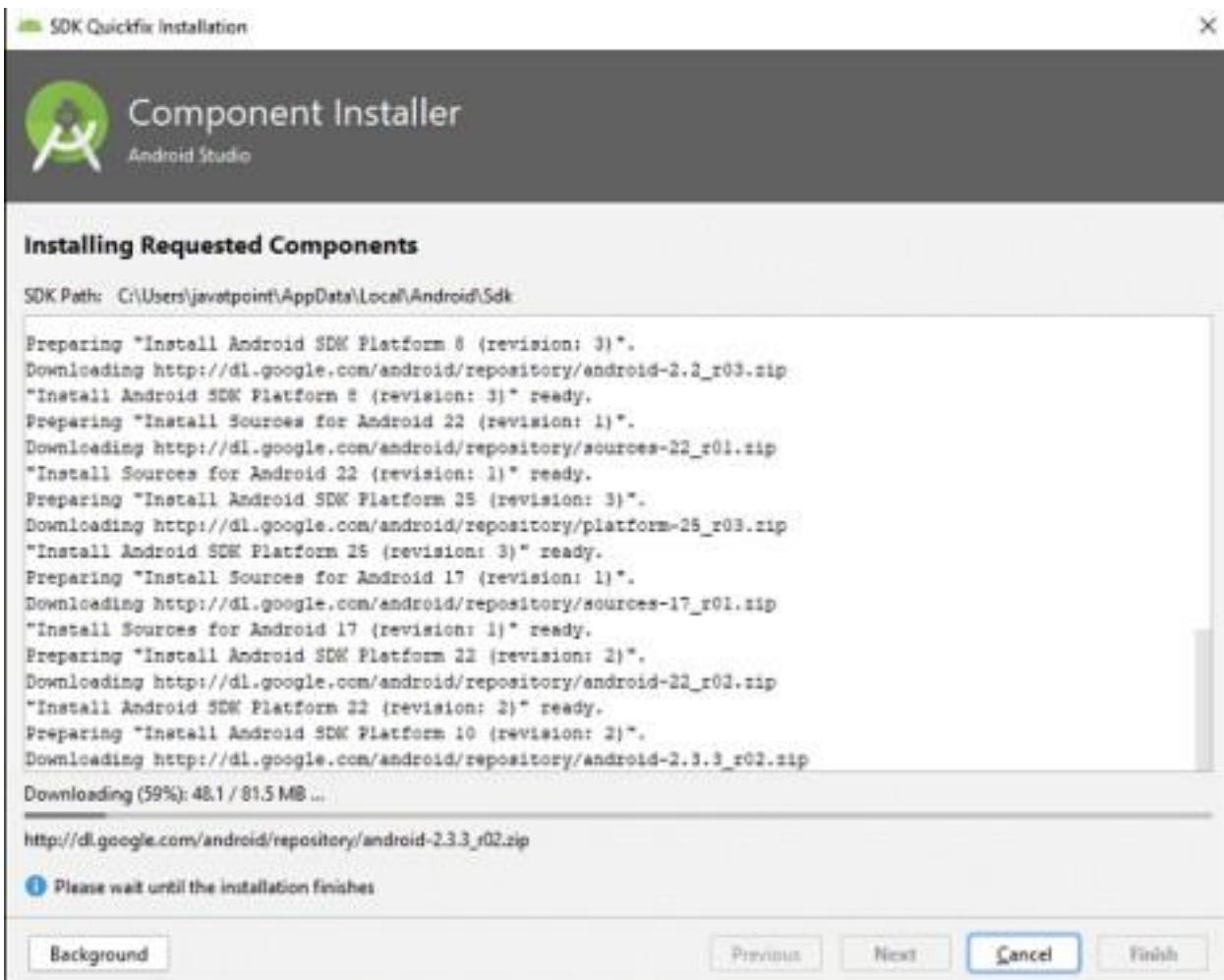
Step 7.2: When the download is complete, open the .exe file and run it. You will get the following dialog box.



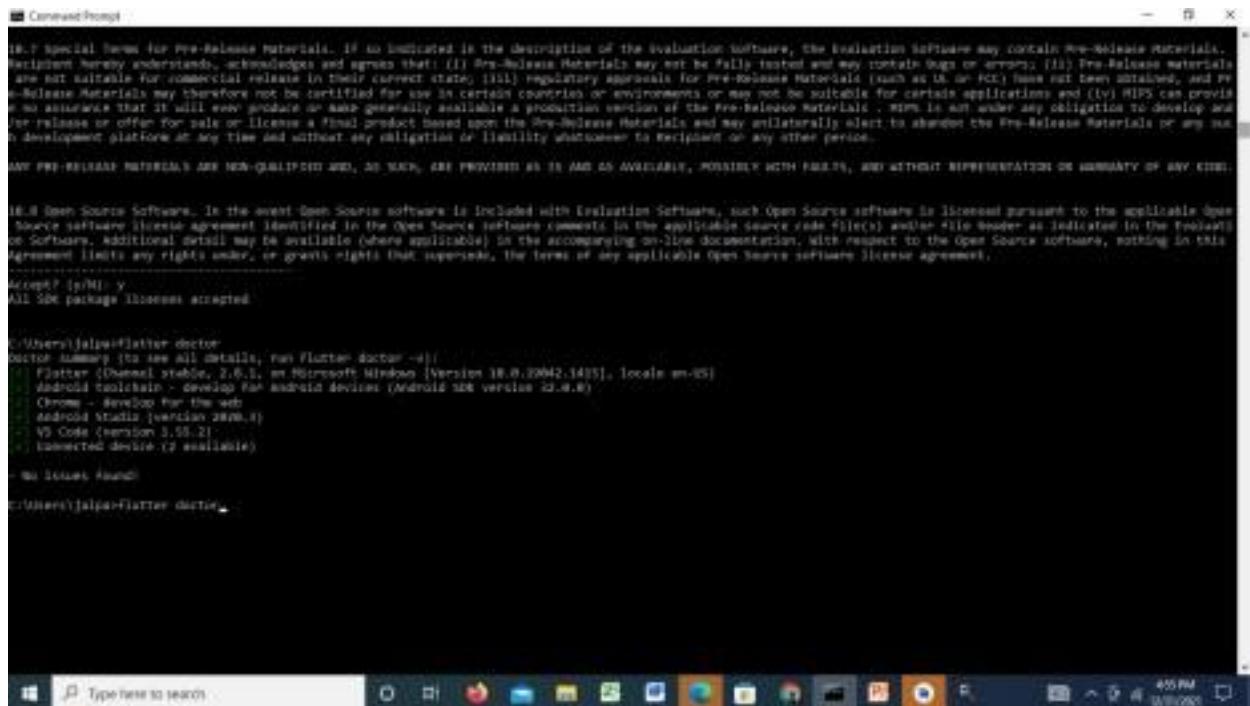
Step 7.3: Follow the steps of the installation wizard. Once the installation wizard completes, you will get the following screen.



Step 7.4: In the above screen, click Next-> Finish. Once the Finish button is clicked, you need to choose the '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.



```
Command Prompt
M&P Special Terms for Pre-Release Materials. If so indicated in the description of the evaluation software, the evaluation software may contain Pre-Release Materials. Participant hereby understands, acknowledges and agrees that: (i) Pre-Release Materials may not be fully tested and may contain bugs or errors; (ii) Pre-Release materials are not suitable for commercial release in their current state; (iii) regulatory approvals for Pre-Release Materials (such as UL or FCC) have not 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) M&P can provide no assurance that it will ever produce or make generally available a production version of the Pre-Release Materials. M&P 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 liability whatsoever to Participant or any other person.

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

M&P Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable open source software license agreement identified in the Open Source license comment 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 software, nothing in this agreement limits any rights under, or grants rights that supersede, the terms of any applicable open source software license agreement.

acceptP [y/N]: y
All 50+ package licenses accepted.

C:\Users\jaiya\Flutter\doctor
doctor summary (to see all details, run flutter doctor -v)
[!] Flutter (Channel stable, 2.6.1, on Microsoft Windows [Version 10.0.20942.1405], locale en-US)
  Android toolchain - develop for Android devices (Android NDK version 23.0.8)
  Chrome - Develop for the web
  Android Studio (version 2020.4)
  VS Code (version 1.55.2)
  Connected device: (2 available)

No issues found!
C:\Users\jaiya\Flutter\doctor
```

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.



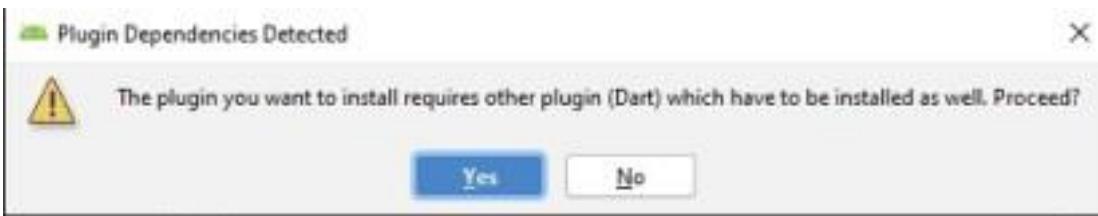
Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.



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.

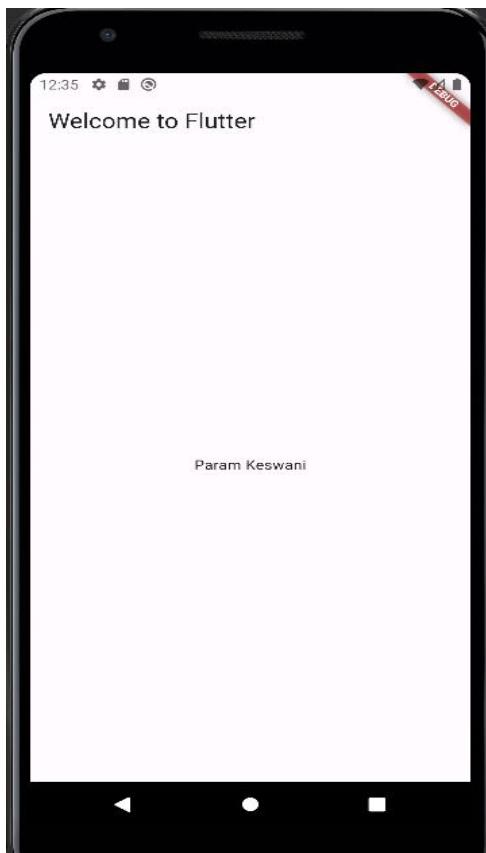


Step 9.3: Restart the Android Studio.

Code:

```
import 'package:flutter/material.dart';
void main() {
  runApp(const MyApp());
}
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: const Text('Welcome to Flutter'),
        ),
        body: const Center(
          child: Text('Param Keswani'),
        ),
      );
  }
}
```

Output:



Conclusion: Hence We ran a simple program on running a simple text, on flutter, running on a virtual device