EXPERIMENT NO.1

Experiment No 1 **1: Installation and Configuration of Flutter Environment.**	
ROLL NO	**29**
NAME	**Payoshni Upendra Khekale**
CLASS	**D15-B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPE D	

Theory:

Flutter, a UI software development toolkit developed by Google, excels in creating natively compiled applications for mobile, web, and desktop from a unified codebase. Explore key aspects that set Flutter apart:

- Unified Codebase:
  Developers leverage Flutter to write code once and deploy it across iOS, Android, web, and desktop platforms seamlessly.

- Dynamic Hot Reload:
  A standout feature, Flutter's Hot Reload allows developers to see instant results of code changes without requiring app restarts, speeding up the development cycle significantly.

- Extensive Widget Library:
  Flutter offers a versatile set of pre-designed and customizable widgets, serving as the foundation for creating modern and responsive user interfaces.

- Artistic UI:
  Flutter supports expressive and flexible UI designs with various animations for a consistent user experience across platforms.

- Dart Programming Language:
  Flutter relies on Dart, a language designed for building modern web and mobile applications, featuring strong typing and just-in-time compilation.

- Tailored Customization and Theming:
  Flutter allows high customization and theming for visually appealing and brand-consistent interfaces.

- Adaptable State Management:
  Flutter's flexible state management system caters to different preferences and project requirements.

- Native Features Accessibility:
  Developers can seamlessly access native features and APIs with Flutter, ensuring integration of platform-specific functionalities.

- Firebase Integration:
  Flutter seamlessly integrates with Firebase, enhancing app functionality with backend services like authentication and database support.

- Web and Desktop Reach:
  In addition to mobile app development, Flutter extends support to web and desktop applications, offering a unified codebase for a broader reach.

- Optimized Performance:
  Flutter compiles apps into native ARM code, optimizing performance through its architecture and the Skia graphics engine.

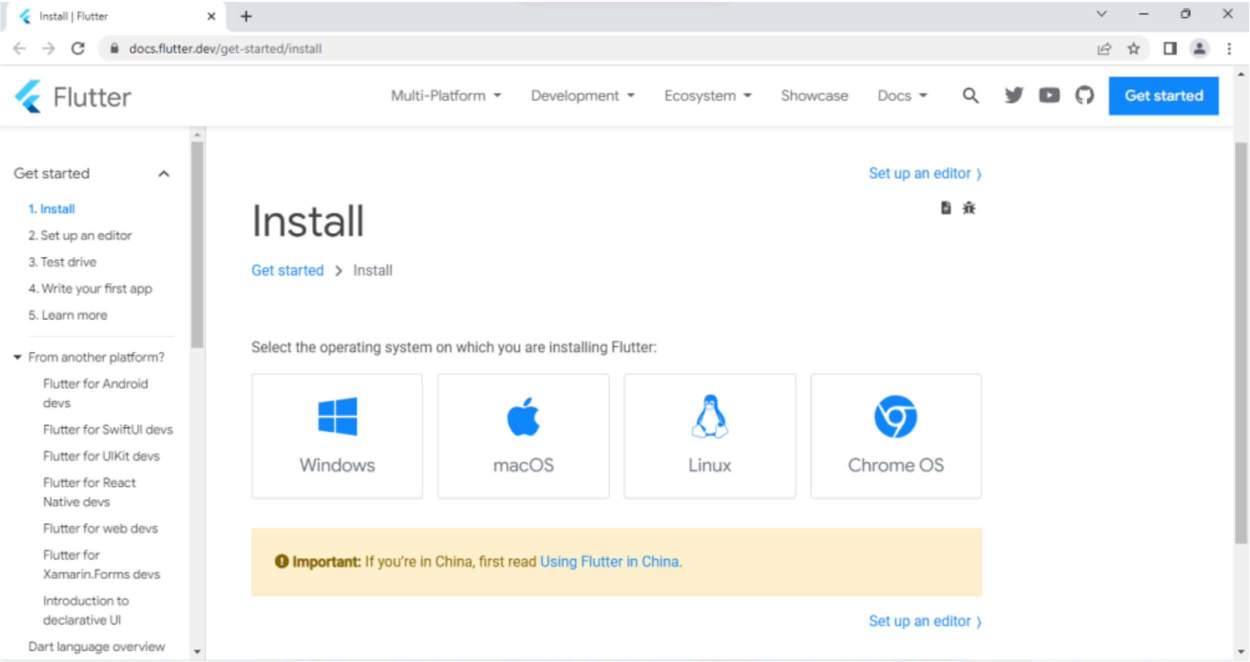
- Expansive Ecosystem:
  Flutter's growing ecosystem on pub.dev provides developers with a range of packages and plugins.

- Engaged Community Support:
  Flutter's vibrant community contributes to its growth, offering support through forums, social media, and organizing events.

These features collectively position Flutter as an ideal choice for developers seeking a versatile and efficient development experience for creating high-performance applications.

Flutter Installation:

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



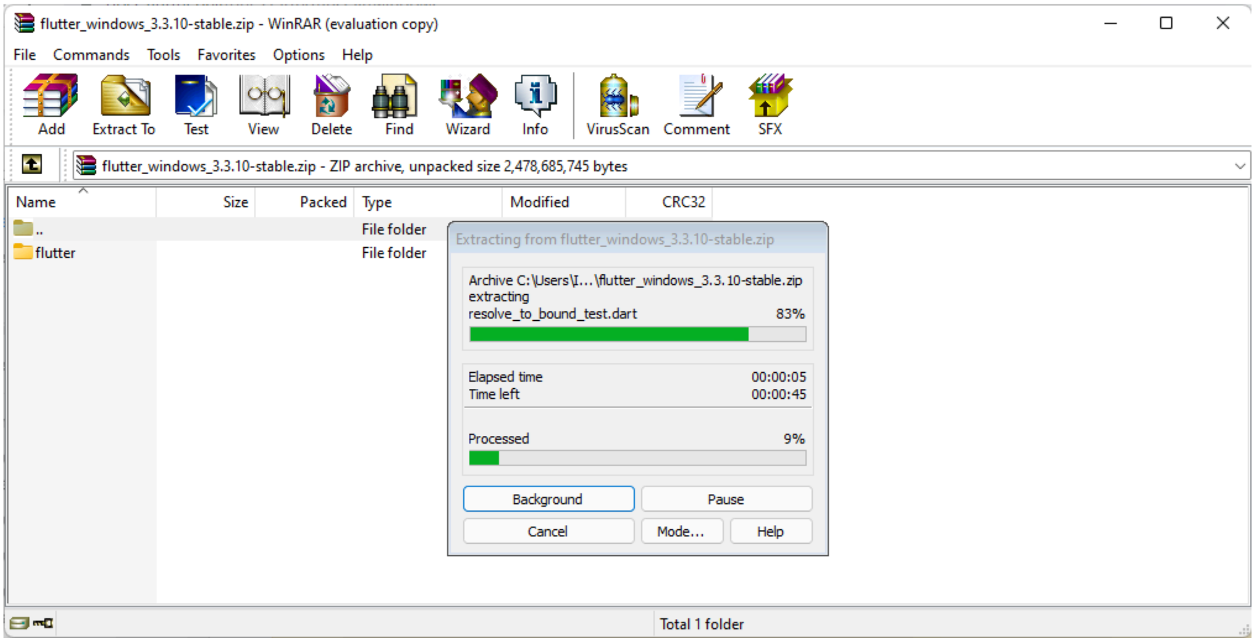
Install

Get started > Install

Select the operating system on which you are installing Flutter:

Important: If you're in China, first read Using Flutter in China.

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



flutter_windows_3.3.10-stable.zip - WinRAR (evaluation copy)

flutter_windows_3.3.10-stable.zip - ZIP archive, unpacked size 2,478,685,745 bytes

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
flutter			File folder		

Extracting from flutter_windows_3.3.10-stable.zip

Archive C:\Users\I...\flutter_windows_3.3.10-stable.zip
extracting
resolve_to_bound_test.dart 83%

Elapsed time 00:00:05
Time left 00:00:45

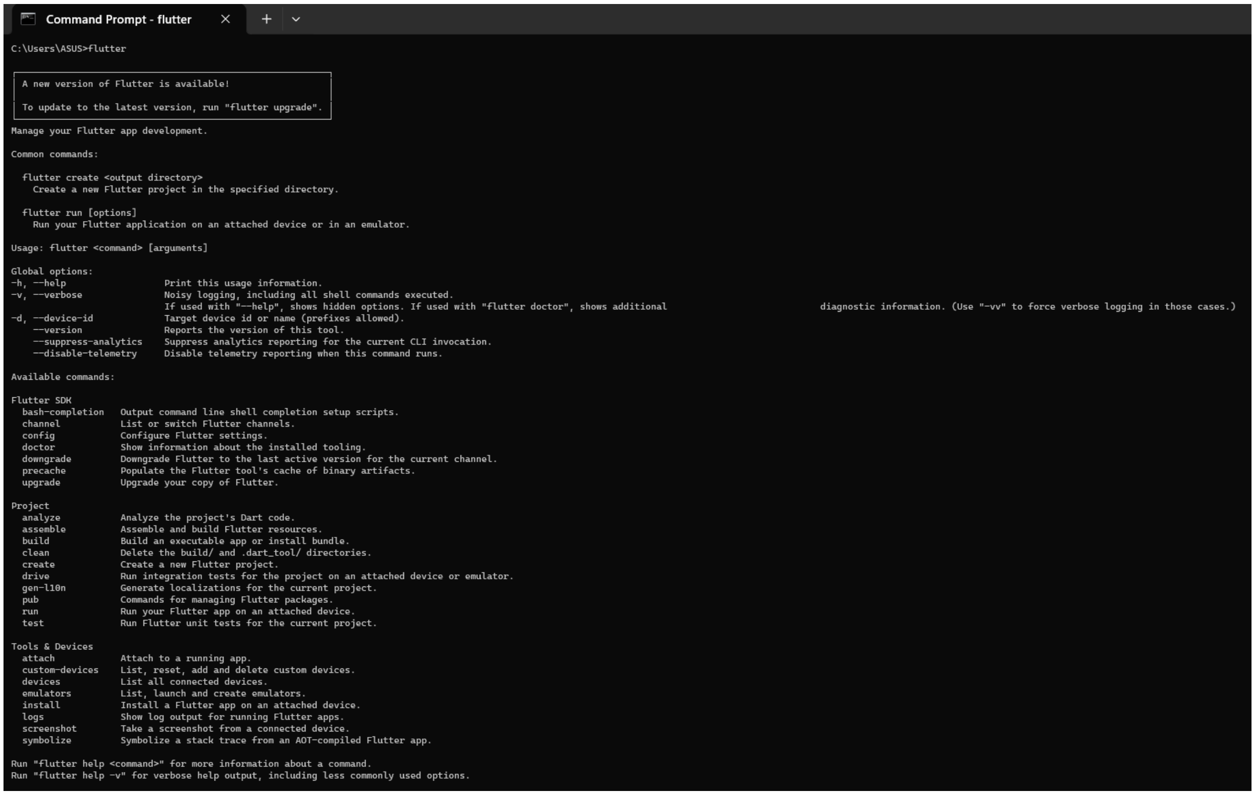
Processed 9%

Total 1 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.

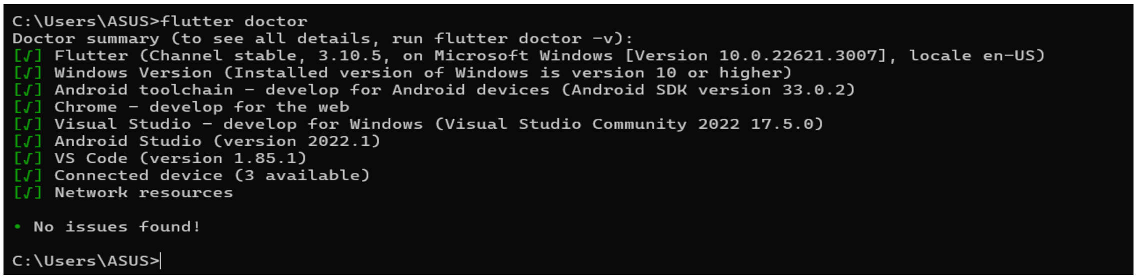
Step 4: Select the Path -> click on Edit and add the path to the Flutter bin folder.

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



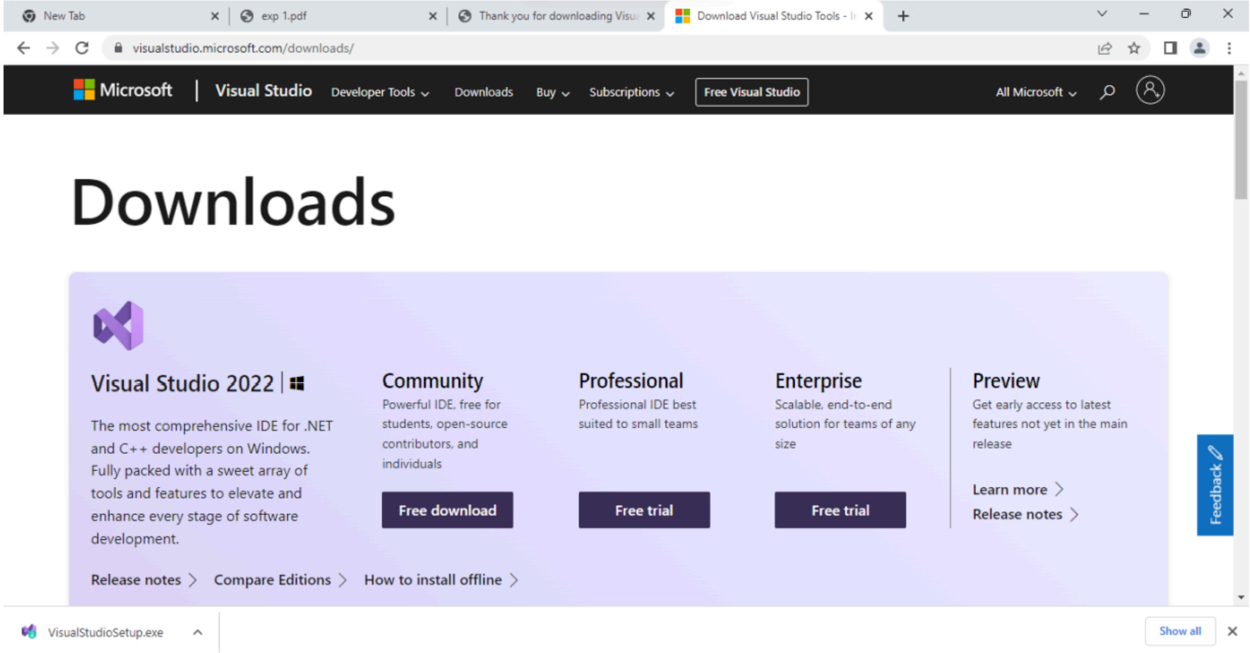
```
C:\Users\ASUS>flutter

 ┌─────────────────────────────────────────────────────────┐
 │ A new version of Flutter is available!                  │
 │                                                         │
 │ To update to the latest version, run "flutter upgrade". │
 └─────────────────────────────────────────────────────────┘
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.
                            If used with "--help", shows hidden options. If used with "flutter doctor", shows additional                            diagnostic information. (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 for the current CLI invocation.
    --disable-telemetry     Disable telemetry reporting when this command runs.

Available commands:

Flutter SDK
  bash-completion   Output command line shell completion setup scripts.
  channel           List or switch Flutter channels.
  config            Configure Flutter settings.
  doctor            Show information about the installed tooling.
  downgrade         Downgrade 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.
  drive             Run integration tests for the project on an attached device or emulator.
  gen-l10n          Generate localizations for the current project.
  pub               Commands for managing Flutter packages.
  run               Run your Flutter app on an attached device.
  test              Run Flutter unit tests for the current project.

Tools & Devices
  attach            Attach to a running app.
  custom-devices    List, reset, add and delete custom devices.
  devices           List all connected devices.
  emulators         List, launch and create emulators.
  install           Install a Flutter app on an attached device.
  logs              Show log output for running Flutter apps.
  screenshot        Take a screenshot from a connected device.
  symbolize         Symbolize a stack trace from an AOT-compiled Flutter app.

Run "flutter help <command>" for more information about a command.
Run "flutter help -v" for verbose help output, including less commonly used options.
```

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.



```
C:\Users\ASUS>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.10.5, on Microsoft Windows [Version 10.0.22621.3007], locale en-US)
[√] 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 for Windows (Visual Studio Community 2022 17.5.0)
[√] Android Studio (version 2022.1)
[√] VS Code (version 1.85.1)
[√] Connected device (3 available)
[√] Network resources

• No issues found!

C:\Users\ASUS>
```

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.



Downloads

Visual Studio 2022

The most comprehensive IDE for .NET and C++ developers on Windows. Fully packed with a sweet array of tools and features to elevate and enhance every stage of software development.

Community
Powerful IDE, free for students, open-source contributors, and individuals

Free download

Professional
Professional IDE best suited to small teams

Free trial

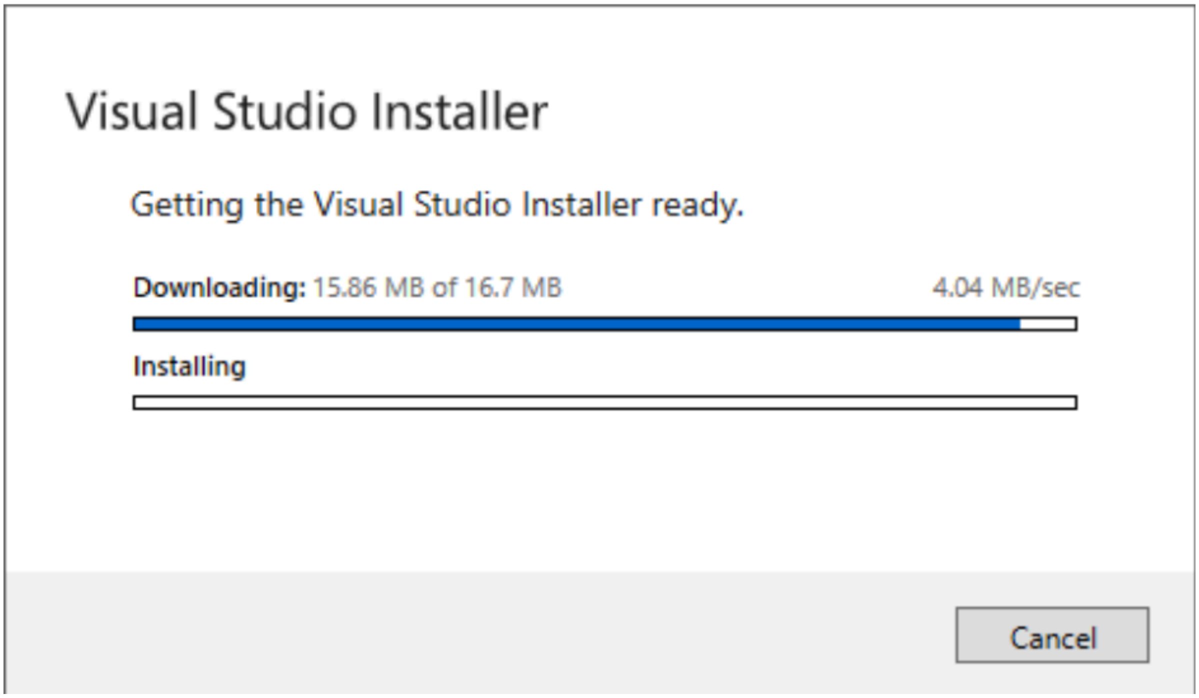
Enterprise
Scalable, end-to-end solution for teams of any size

Free trial

Preview
Get early access to latest features not yet in the main release

Learn more
Release notes

Release notes   Compare Editions   How to install offline



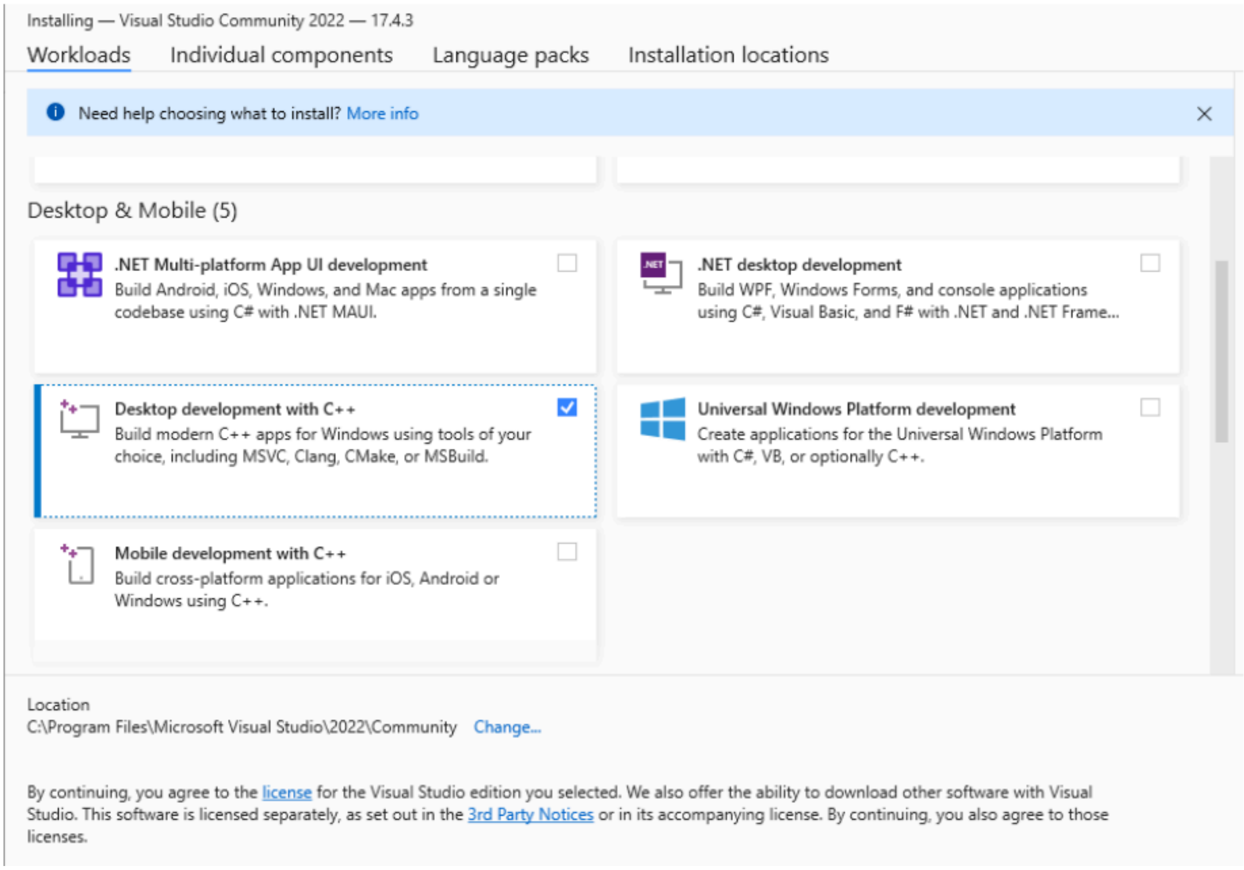
Visual Studio Installer

Getting the Visual Studio Installer ready.

Downloading: 15.86 MB of 16.7 MB   4.04 MB/sec

Installing

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



Need help choosing what to install? More info

Desktop & Mobile (5)

□ .NET Multi-platform App UI development
Build Android, iOS, Windows, and Mac apps from a single codebase using C# with .NET MAUI.

□ .NET desktop development
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F# with .NET and .NET Frame...

☑ Desktop development with C++
Build modern C++ apps for Windows using tools of your choice, including MSVC, Clang, CMake, or MSBuild.

□ Universal Windows Platform development
Create applications for the Universal Windows Platform with C#, VB, or optionally C++.

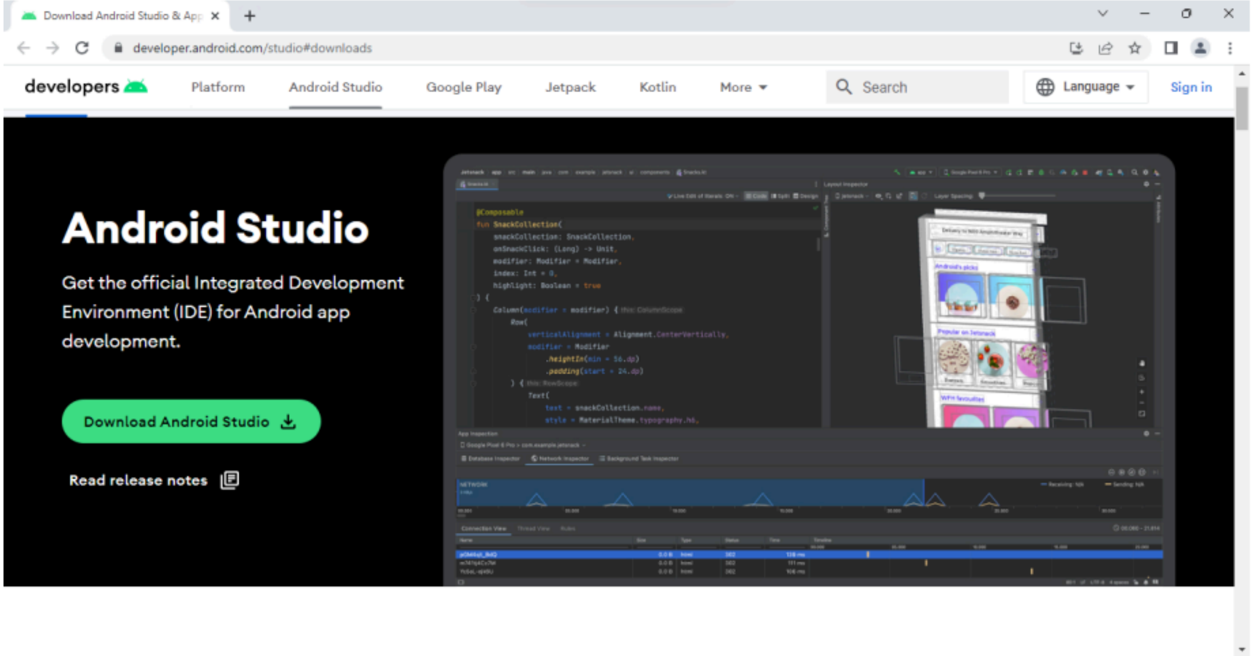
□ Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

Location
C:\Program Files\Microsoft Visual Studio\2022\Community

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

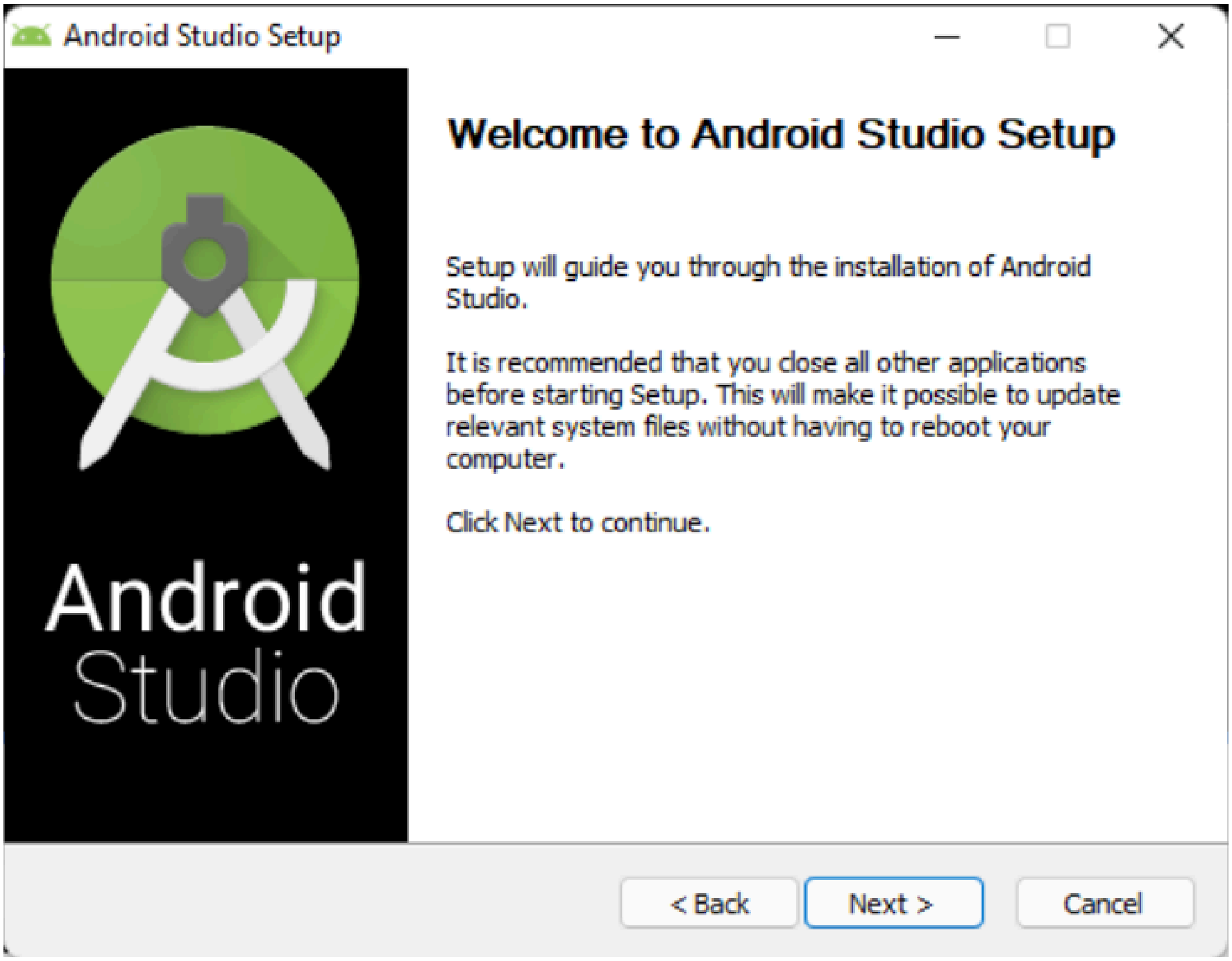
We can see that the Visual Studio issue has been solved.

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.



Android Studio

Get the official Integrated Development Environment (IDE) for Android app development.



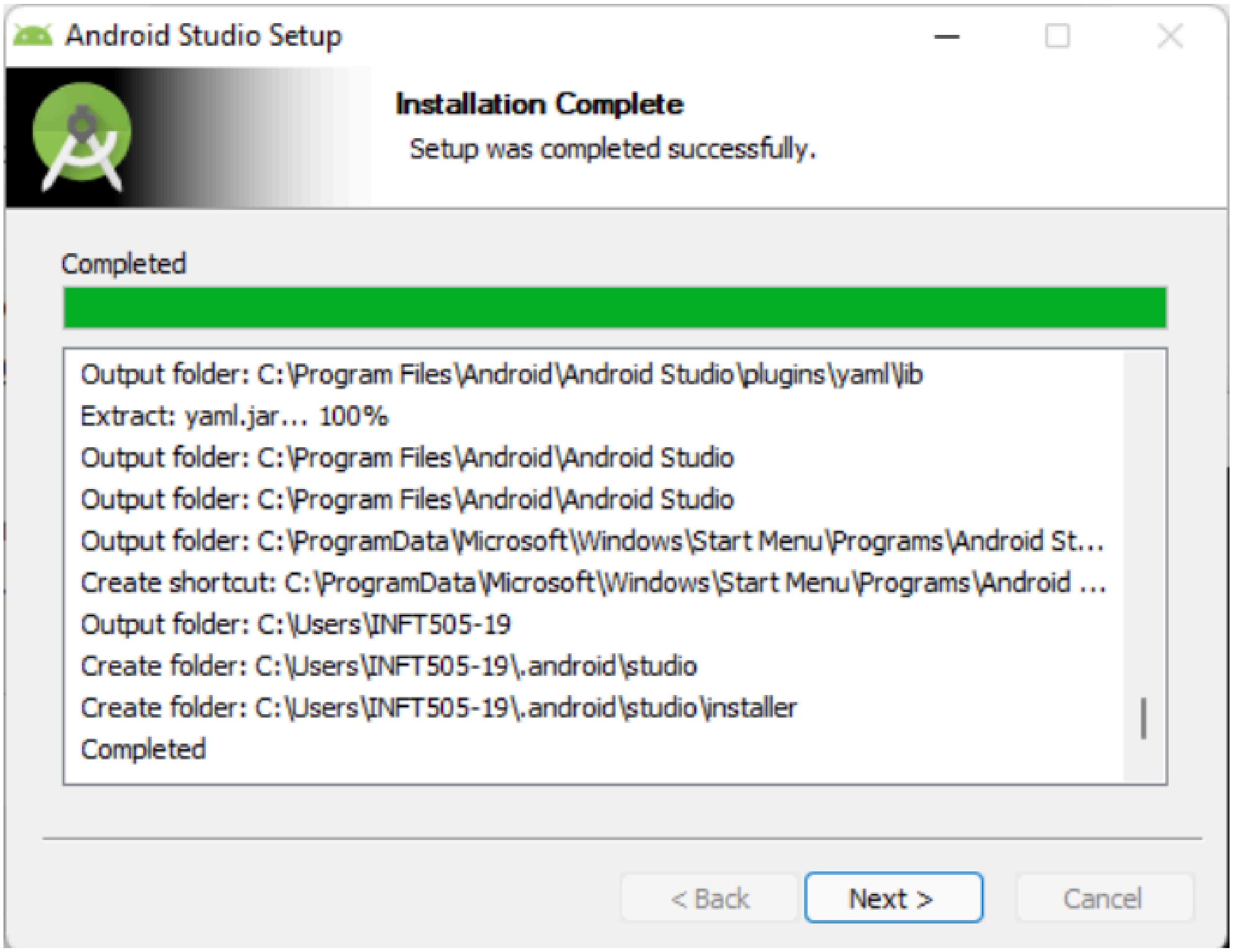
Android Studio Setup

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.



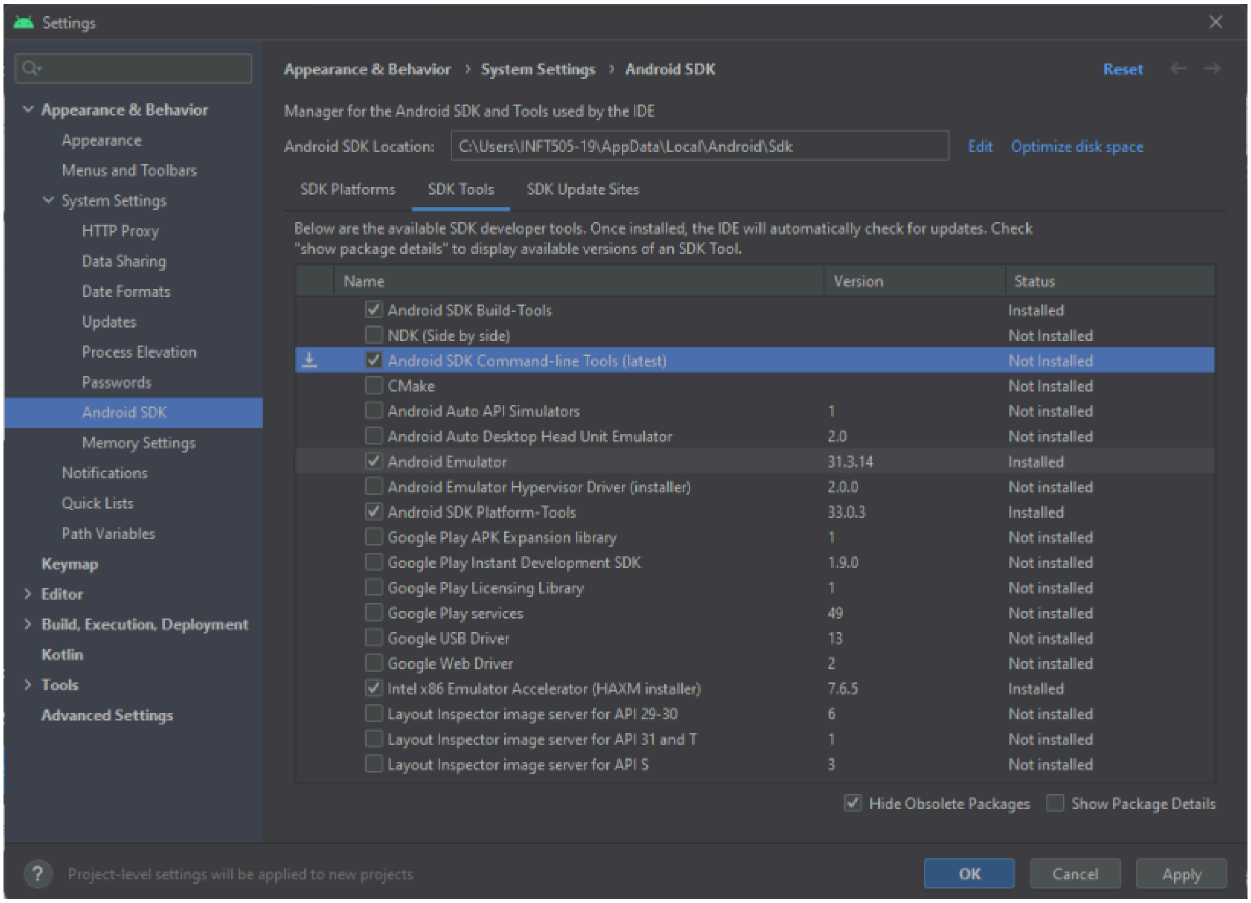
Installation Complete

Setup was completed successfully.

Completed

```
Output folder: C:\Program Files\Android\Android Studio\plugins\yaml\lib
Extract: yaml.jar... 100%
Output folder: C:\Program Files\Android\Android Studio
Output folder: C:\Program Files\Android\Android Studio
Output folder: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Android St...
Create shortcut: C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Android ...
Output folder: C:\Users\INFT505-19
Create folder: C:\Users\INFT505-19\.android\studio
Create folder: C:\Users\INFT505-19\.android\studio\installer
Completed
```

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



Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\INFT505-19\AppData\Local\Android\Sdk

Below are the available SDK developer tools. Once installed, the IDE will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

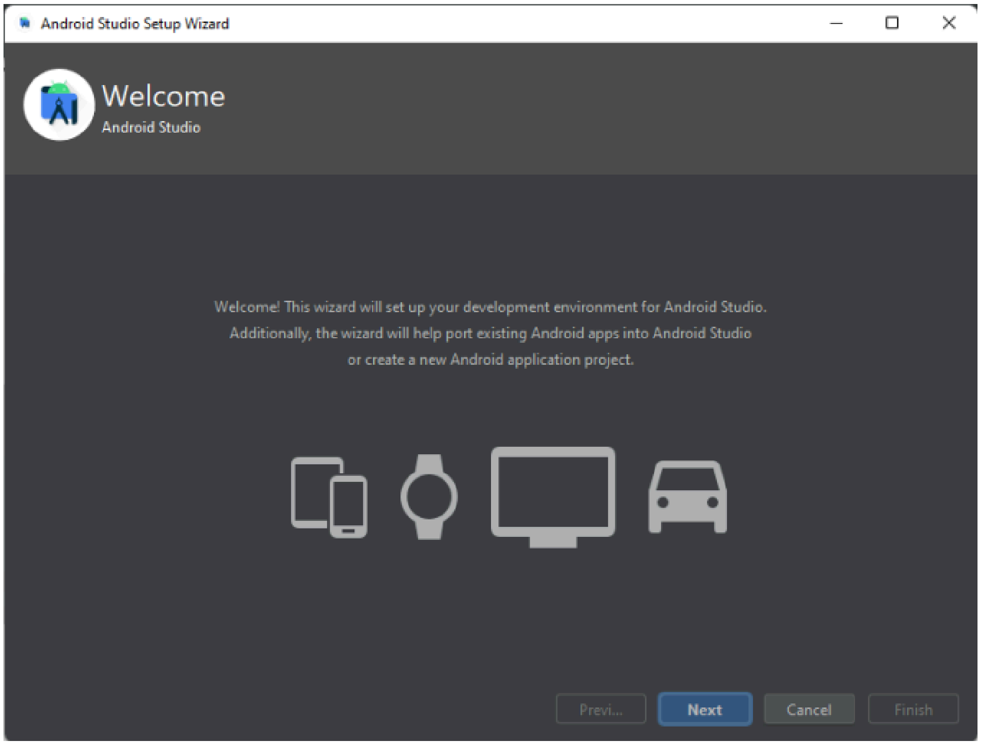
Name	Version	Status
☑ Android SDK Build-Tools		Installed
□ NDK (Side by side)		Not Installed
☑ Android SDK Command-line Tools (latest)		Not Installed
□ CMake		Not Installed
□ Android Auto API Simulators	1	Not installed
□ Android Auto Desktop Head Unit Emulator	2.0	Not installed
☑ Android Emulator	31.3.14	Installed
□ Android Emulator Hypervisor Driver (installer)	2.0.0	Not installed
☑ Android SDK Platform-Tools	33.0.3	Installed
□ Google Play APK Expansion library	1	Not installed
□ Google Play Instant Development SDK	1.9.0	Not installed
□ Google Play Licensing Library	1	Not installed
□ Google Play services	49	Not installed
□ Google USB Driver	13	Not installed
□ Google Web Driver	2	Not installed
☑ Intel x86 Emulator Accelerator (HAXM installer)	7.6.5	Installed
□ Layout Inspector image server for API 29-30	6	Not installed
□ Layout Inspector image server for API 31 and T	1	Not installed
□ Layout Inspector image server for API S	3	Not installed

☑ Hide Obsolete Packages □ Show Package Details

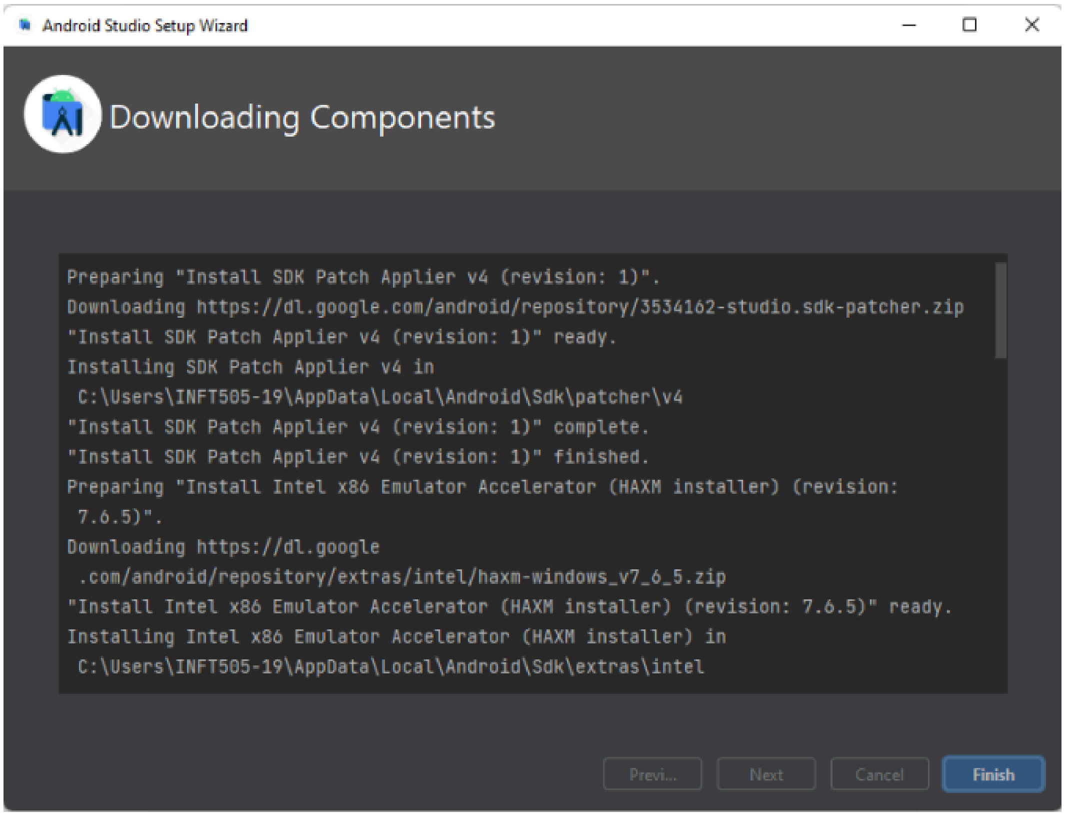
Project-level settings will be applied to new projects

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.

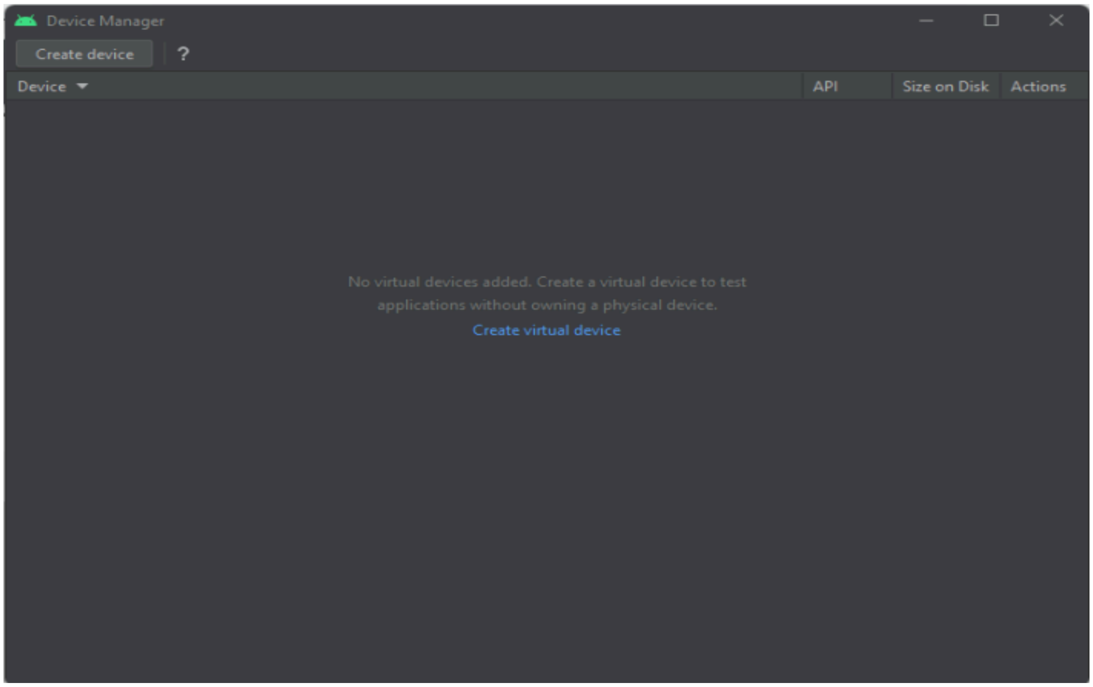


Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.



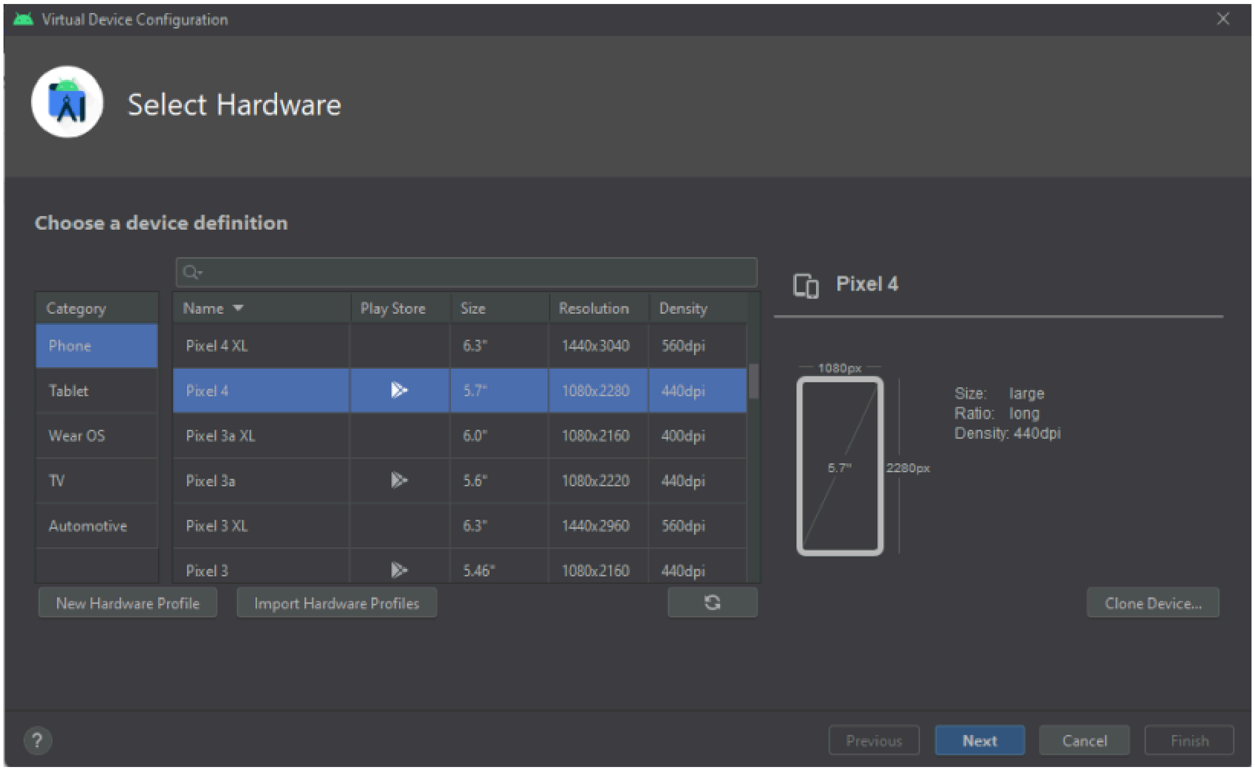
```
Preparing "Install SDK Patch Applier v4 (revision: 1)".
Downloading https://dl.google.com/android/repository/3534162-studio.sdk-patcher.zip
"Install SDK Patch Applier v4 (revision: 1)" ready.
Installing SDK Patch Applier v4 in
 C:\Users\INFT505-19\AppData\Local\Android\Sdk\patcher\v4
"Install SDK Patch Applier v4 (revision: 1)" complete.
"Install SDK Patch Applier v4 (revision: 1)" finished.
Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision:
 7.6.5)".
Downloading https://dl.google
 .com/android/repository/extras/intel/haxm-windows_v7_6_5.zip
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 7.6.5)" ready.
Installing Intel x86 Emulator Accelerator (HAXM installer) in
 C:\Users\INFT505-19\AppData\Local\Android\Sdk\extras\intel
```

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:



No virtual devices added. Create a virtual device to test applications without owning a physical device.

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

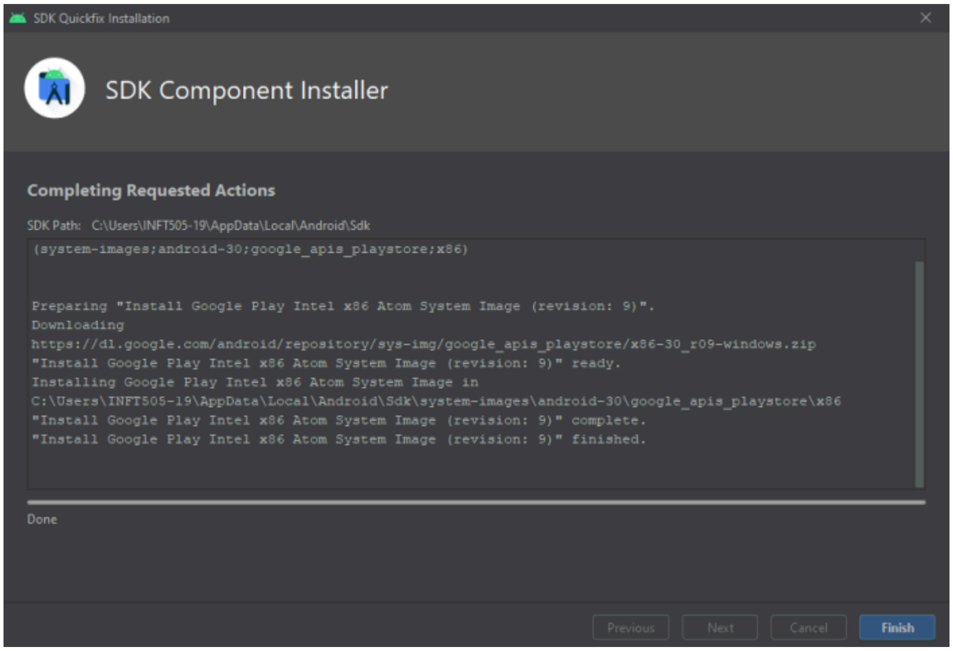


Choose a device definition

Name	Play Store	Size	Resolution	Density
Pixel 4 XL		6.3"	1440x3040	560dpi
Pixel 4	▶	5.7"	1080x2280	440dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi

Size: large
Ratio: long
Density: 440dpi

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



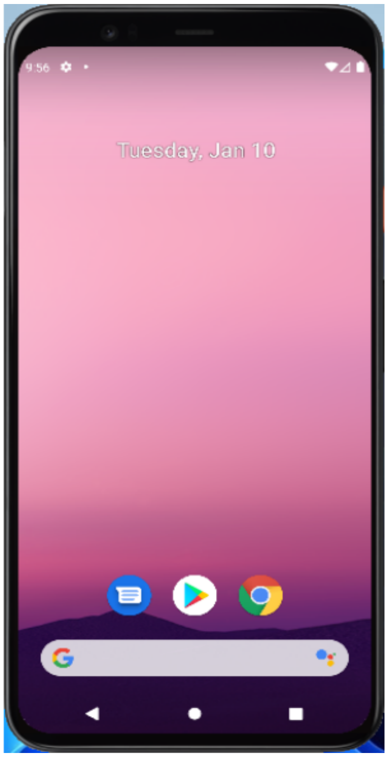
Completing Requested Actions

SDK Path: C:\Users\INFT505-19\AppData\Local\Android\Sdk

```
(system-images;android-30;google_apis_playstore;x86)


Preparing "Install Google Play Intel x86 Atom System Image (revision: 9)".
Downloading
https://dl.google.com/android/repository/sys-img/google_apis_playstore/x86-30_r09-windows.zip
"Install Google Play Intel x86 Atom System Image (revision: 9)" ready.
Installing Google Play Intel x86 Atom System Image in
C:\Users\INFT505-19\AppData\Local\Android\Sdk\system-images\android-30\google_apis_playstore\x86
"Install Google Play Intel x86 Atom System Image (revision: 9)" complete.
"Install Google Play Intel x86 Atom System Image (revision: 9)" finished.
```

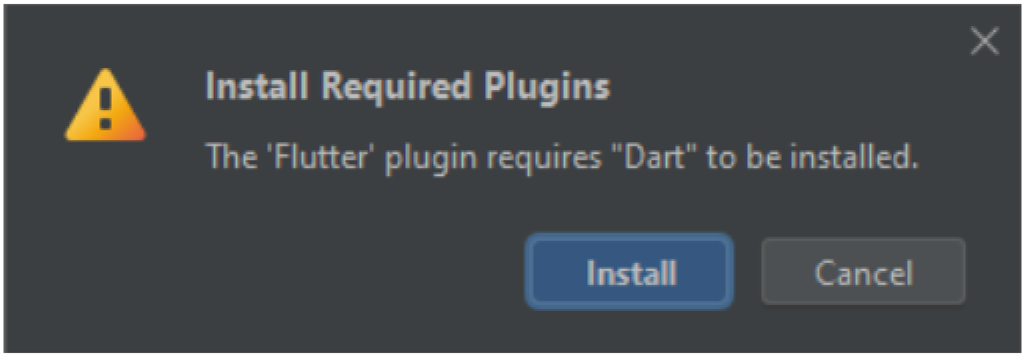
Done

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.



Install Required Plugins

The 'Flutter' plugin requires "Dart" to be installed.

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

\

Conclusion:

To Conclude,the smooth installation of Flutter during this lab experiment not only underscores its user-friendly setup but also highlights its adaptability. This versatile framework not only simplifies the development process but also serves as a catalyst for innovation in the realm of cross-platform applications, enhancing the landscape of mobile app development.