EXPERIMENT NO 1

Aim: To install and configure the Flutter Environment

Theory:

Flutter: A Powerful Framework for Cross-Platform Development
Flutter is an open-source UI software development framework created by Google. It enables developers to build visually appealing, natively compiled applications for mobile, web, desktop, and embedded platforms — all from a single codebase.

Key Features of Flutter:

- High Performance:
  Flutter compiles directly to ARM or Intel machine code, as well as JavaScript for web, ensuring fast and smooth performance across all devices and platforms.

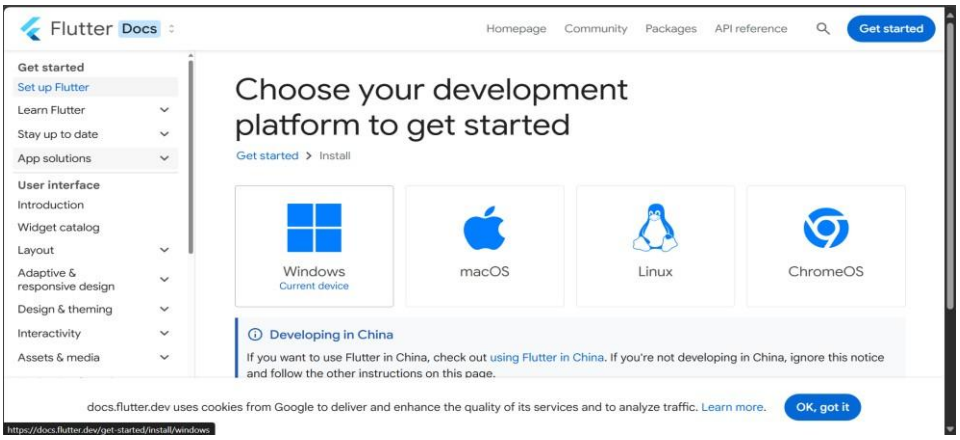
- Developer Productivity:
  With Flutter's Hot Reload feature, developers can instantly view code changes without restarting the application or losing its current state, significantly speeding up the development cycle.

- Design Flexibility:
  Flutter offers full control over every pixel on the screen, allowing for highly customized and adaptive UI designs that provide a native look and feel across different screen sizes and device types.

In essence, Flutter empowers developers to create modern, responsive, and feature-rich applications efficiently and consistently across platforms.

Installation of 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/getstarted/install , you will get the following screen.

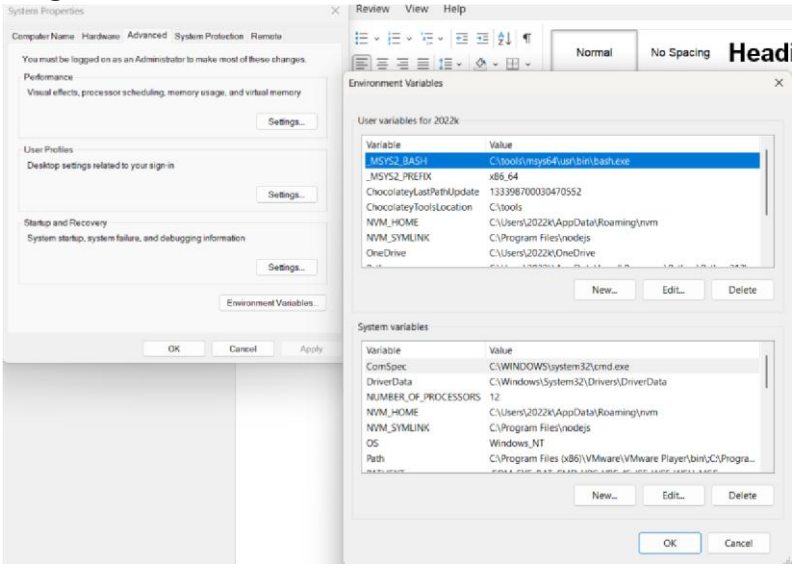


Step 2: Next, to download the latest Flutter SDK, click on the Windows **icon**. Here, you will find the download link for SDK.

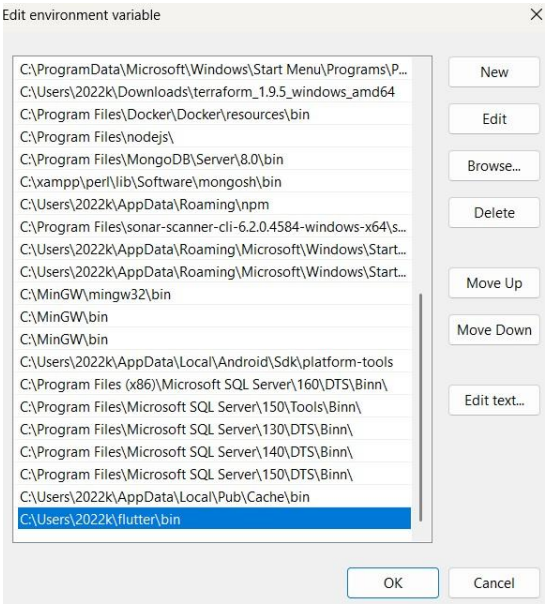
Step 3: When your download is complete, extract the **zip** file and place it in the desired installation folder or location, for example, C: /Flutter.

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.

```
C:\Users\2022k>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.
                            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.
    --enable-analytics      Enable telemetry reporting each time a flutter or dart command runs.
    --disable-analytics     Disable telemetry reporting each time a flutter or dart command runs, until it is
                            re-enabled.
    --suppress-analytics    Suppress analytics reporting for the current CLI invocation.

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             Builds and installs the app, and runs a Dart program that connects to the app, often to run
                    externally facing integration tests, such as with package:test and package:flutter_driver.
  gen-l10n          Generate localizations for the current project.
  pub               Commands for managing Flutter packages.
```

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.

Step 6: Analyze System with flutter doctor

```
C:\Users\Sadneya>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.22.2, on Microsoft Windows [Version 10.0.22631.4751], locale en-IN)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
    X Cannot execute C:\Program Files\Java\jdk-17\bin\java to determine the version
[✓] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Enterprise 2022 17.12.4)
    X Visual Studio is missing necessary components. Please re-run the Visual Studio installer for the "Desktop development with C++" workload, and include these components:
        MSVC v142 - VS 2019 C++ x64/x86 build tools
          - If there are multiple build tool versions available, install the latest
        C++ CMake tools for Windows
        Windows 10 SDK
[!] Android Studio (not installed)
[✓] VS Code (version 1.97.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 3 categories.
```

When you run the flutter doctor command, it analyzes your system and generates a report, as shown in the image below. This report provides details about any missing tools required to run Flutter, as well as development tools that are available but not yet connected to your device.

Android Studio Overview:

Android Studio is the official integrated development environment (IDE) for Android application development, based on JetBrains' IntelliJ IDEA. Specifically tailored for Android development, it replaces the older Eclipse Android Development Tools (E-ADT). Android Studio is available for Windows, macOS, and Linux operating systems.

Installation of Android Studio:

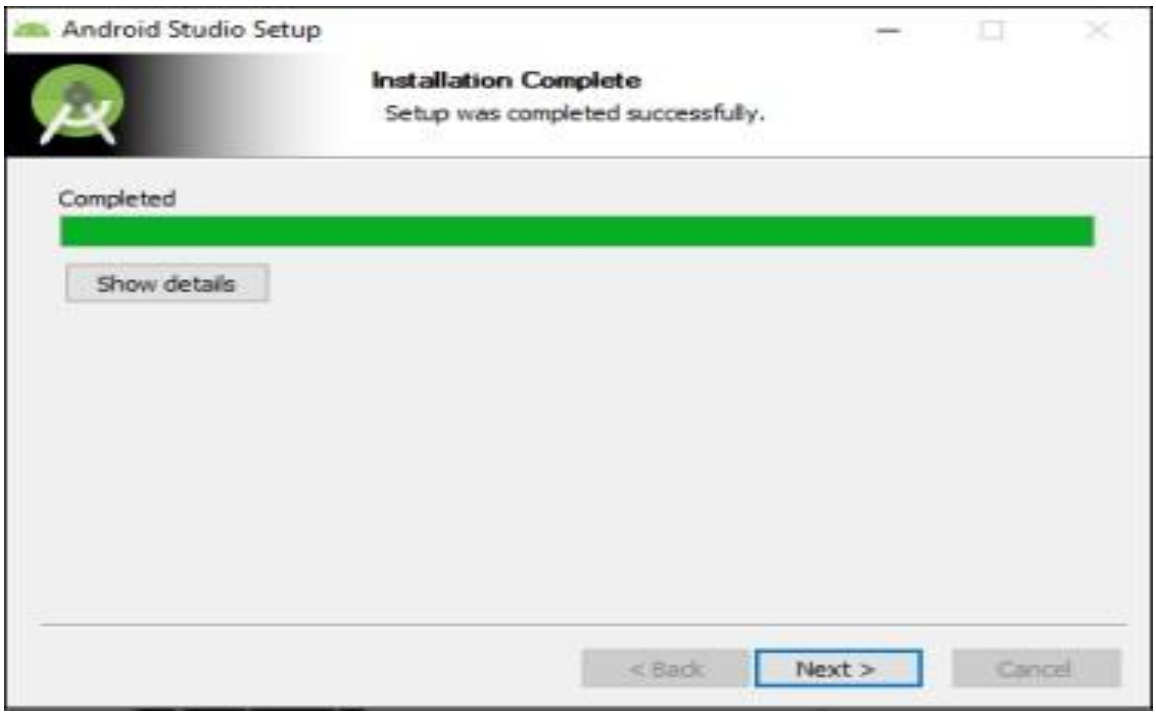
Step 7: Install the Android SDK

If flutter doctor indicates that the Android SDK is missing, the next step is to install Android Studio, which includes the SDK. Follow these steps:

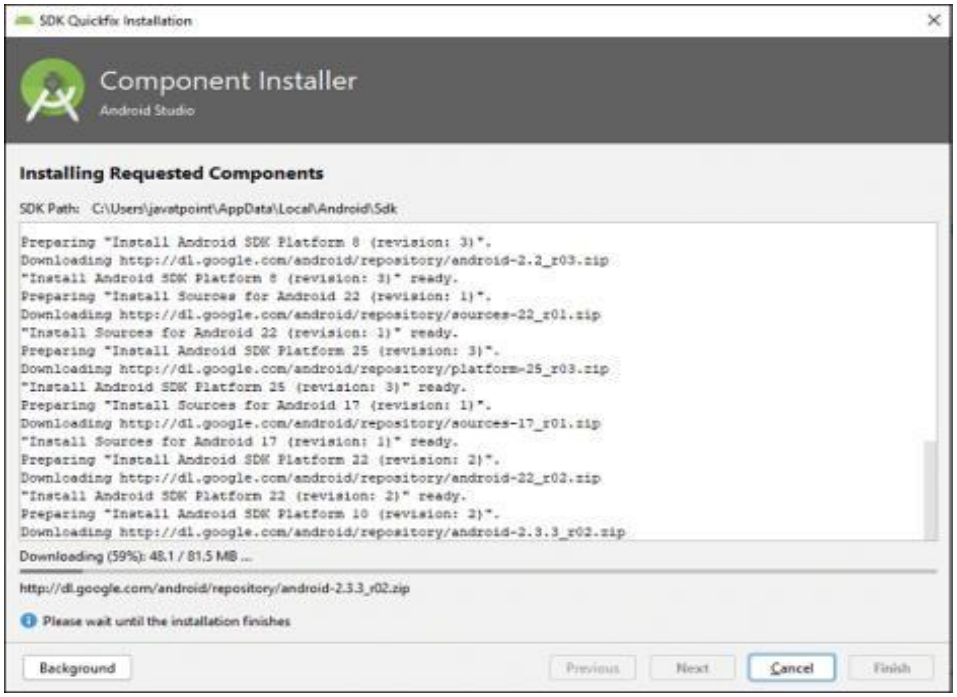
Step 7.1: Download the latest version of Android Studio from the official website.

Step 7.2: Once the download is complete, open the .exe file and run the installer. You will be prompted with a setup dialog to proceed with the installation.

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.

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.

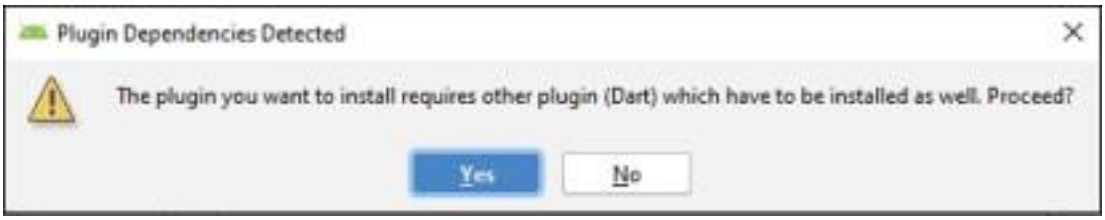
Step 8.5: Last, click on the icon pointed into the red color rectangle. The Android emulator displayed as below screen.

Step 9: Install Flutter and Dart Plugins in Android Studio

To build Flutter applications in Android Studio, you need to install the Flutter and Dart plugins. These plugins provide templates for creating Flutter projects and enable you to run and debug applications directly within the IDE. Follow the steps below to install them:

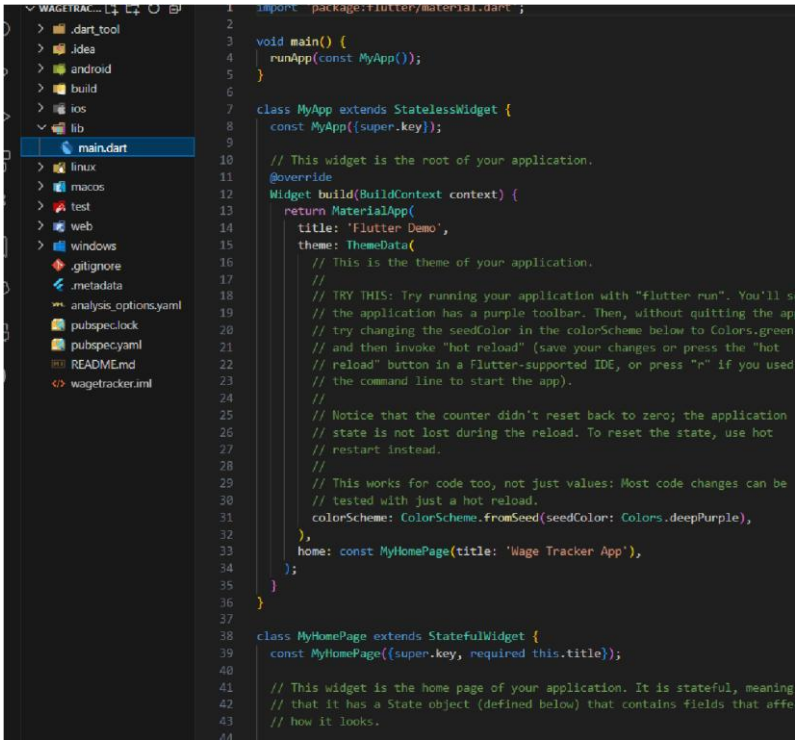
Step 9.1: Launch Android Studio and navigate to File → Settings → Plugins. Step 9.2: In the search bar, type Flutter. Once the Flutter plugin appears, select it and click Install.

During the installation, a prompt will appear asking to install the Dart plugin as well. Click Yes to continue.

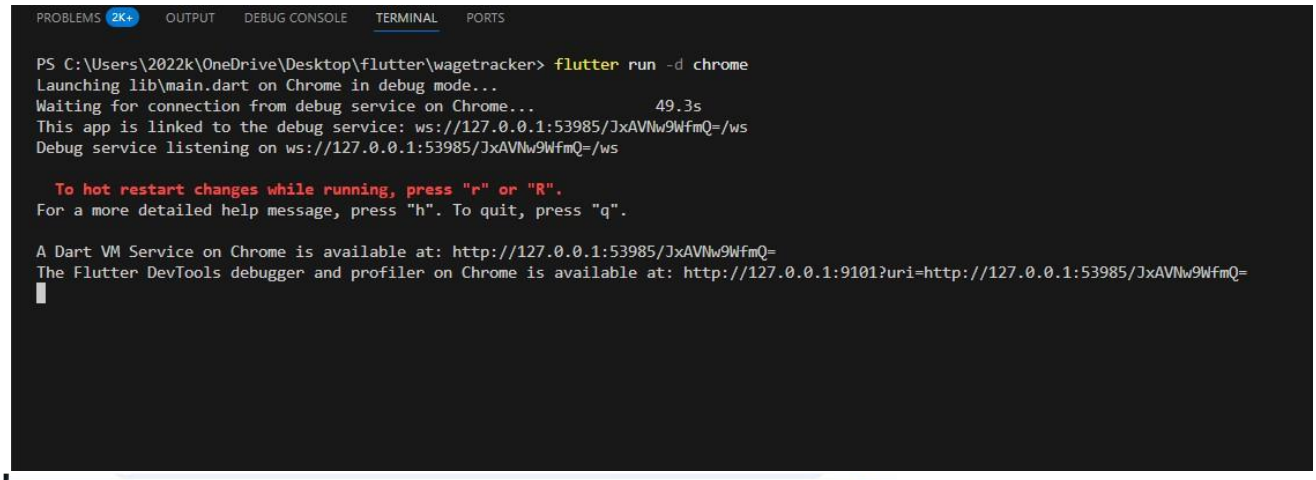


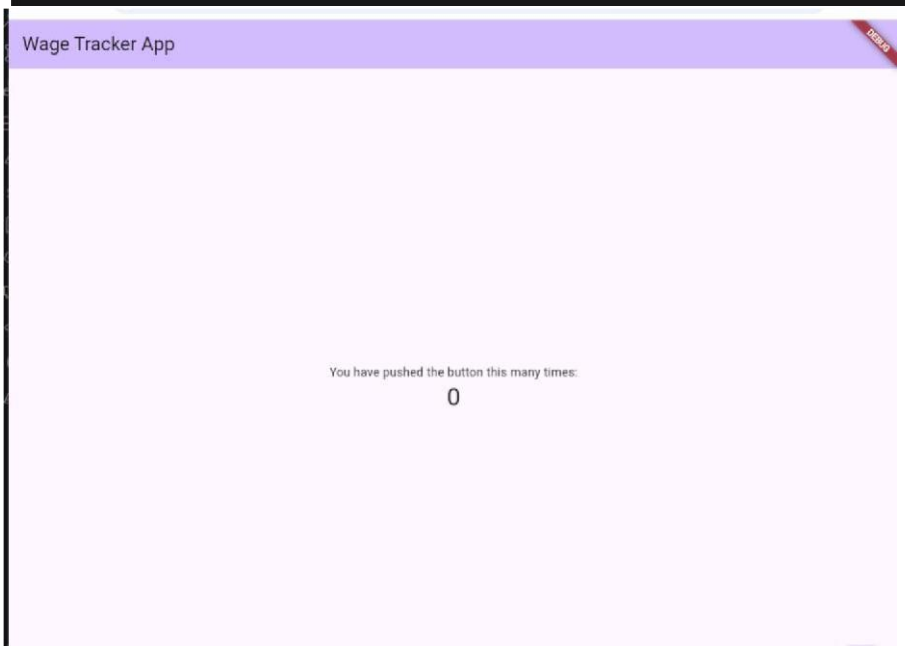
Step 9.3: Restart the Android Studio.**Creation of flutter project:**

1. Create a folder with the name of your project.
2. Then open it with vscode.
3. Then click on the open integrated terminal and type the command: “flutter create .”
4. This will create the project with the folder name.
5. Now enter the code in main.dart.



6. Then in command type “flutter run” to run on emulator and with a “-d chrome” extension we can run it on browser (here, chrome).





.

Conclusion: This experiment outlines the comprehensive setup process for the Flutter development environment. It includes the installation of key components such as the Flutter SDK, Android Studio IDE, and essential plugins. The flutter doctor tool plays a crucial role by detecting and assisting in resolving any missing dependencies. After successful configuration, developers can create and run Flutter projects on emulators or physical devices, laying the groundwork for building cross-platform mobile applications using Flutter.