Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

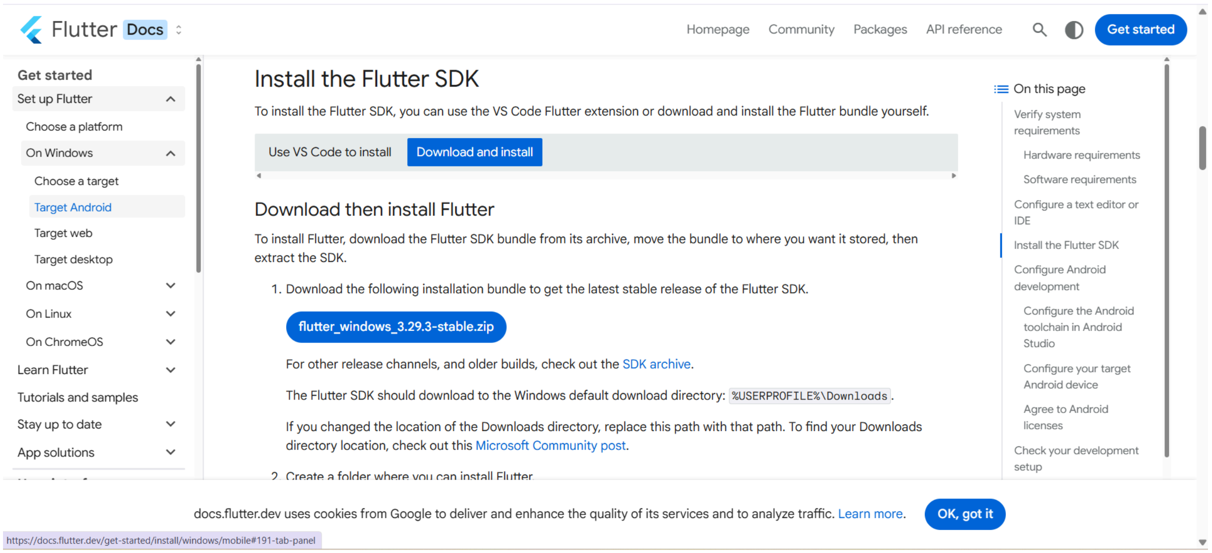
Experiment 1

Aim: Installation and Configuration of Flutter Environment.

Install the Flutter SDK

Step 1: Download the Flutter SDK for Windows

- Visit the official Flutter website at https://docs.flutter.dev/get-started/install. You'll see the Flutter installation page.



Install the Flutter SDK

To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

For other release channels, and older builds, check out the SDK archive.

The Flutter SDK should download to the Windows default download directory: `%USERPROFILE%\Downloads`.

If you changed the location of the Downloads directory, replace this path with that path. To find your Downloads directory location, check out this Microsoft Community post.

2. Create a folder where you can install Flutter.

docs.flutter.dev uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. Learn more.

https://docs.flutter.dev/get-started/install/windows/mobile#191-tab-panel

Step 2: Download the SDK

- Locate and click the Windows icon on the website to access the download link for the latest Flutter SDK.

Step 3: Extract the SDK

- Once the download completes, extract the zip file to a preferred location, such as C:\Flutter.

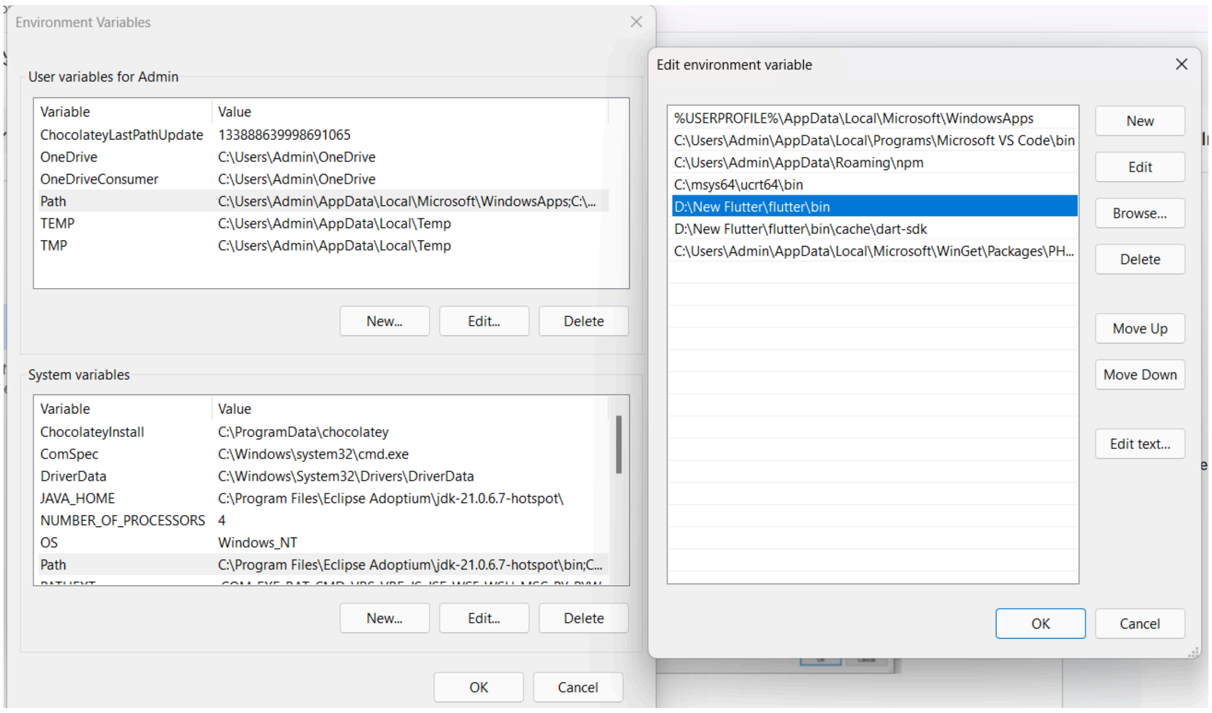
Step 4: Update the System Path

- To run Flutter commands from the Windows Command Prompt, add the Flutter bin directory to your system's PATH environment variable. Follow these steps:

- Step 4.1: Right-click on This PC or My Computer, select Properties, go to the Advanced tab, and click Environment Variables.
- Step 4.2: In the Environment Variables window, find the Path variable under System Variables, select it, and click Edit.
- Step 4.3: Click New, add the path to the Flutter bin folder (e.g., C:\Flutter\bin), then click OK to save all changes.

Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**


Environment Variables

User variables for Admin

Variable	Value
ChocolateyLastPathUpdate	133888639998691065
OneDrive	C:\Users\Admin\OneDrive
OneDriveConsumer	C:\Users\Admin\OneDrive
Path	C:\Users\Admin\AppData\Local\Microsoft\WindowsApps;C:\...
TEMP	C:\Users\Admin\AppData\Local\Temp
TMP	C:\Users\Admin\AppData\Local\Temp

System variables

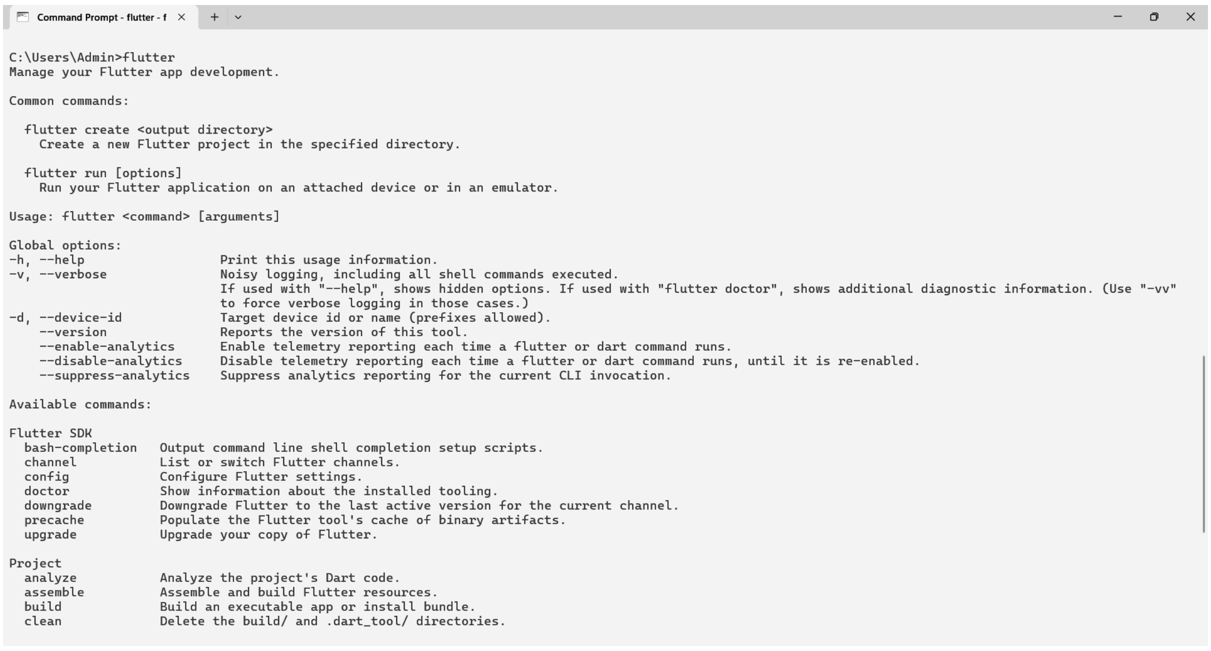
Variable	Value
ChocolateyInstall	C:\ProgramData\chocolatey
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
JAVA_HOME	C:\Program Files\Eclipse Adoptium\jdk-21.0.6.7-hotspot\
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\Program Files\Eclipse Adoptium\jdk-21.0.6.7-hotspot\bin;C...

Edit environment variable

```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\Admin\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\Admin\AppData\Roaming\npm
C:\msys64\ucrt64\bin
D:\New Flutter\flutter\bin
D:\New Flutter\flutter\bin\cache\dart-sdk
C:\Users\Admin\AppData\Local\Microsoft\WinGet\Packages\PH...
```

Step 5: Verify Flutter Installation

- Open a Command Prompt and run the following command:
  flutter
  Next, run:
  flutter doctor
  The flutter doctor command checks your system for Flutter development requirements and generates a report showing the status of your setup.



```
C:\Users\Admin>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.
```

Step 6: Review the Flutter Doctor Report

- After running flutter doctor, you'll see a report detailing any missing tools or configurations needed for Flutter development, as well as the status of installed tools not yet connected to a device.

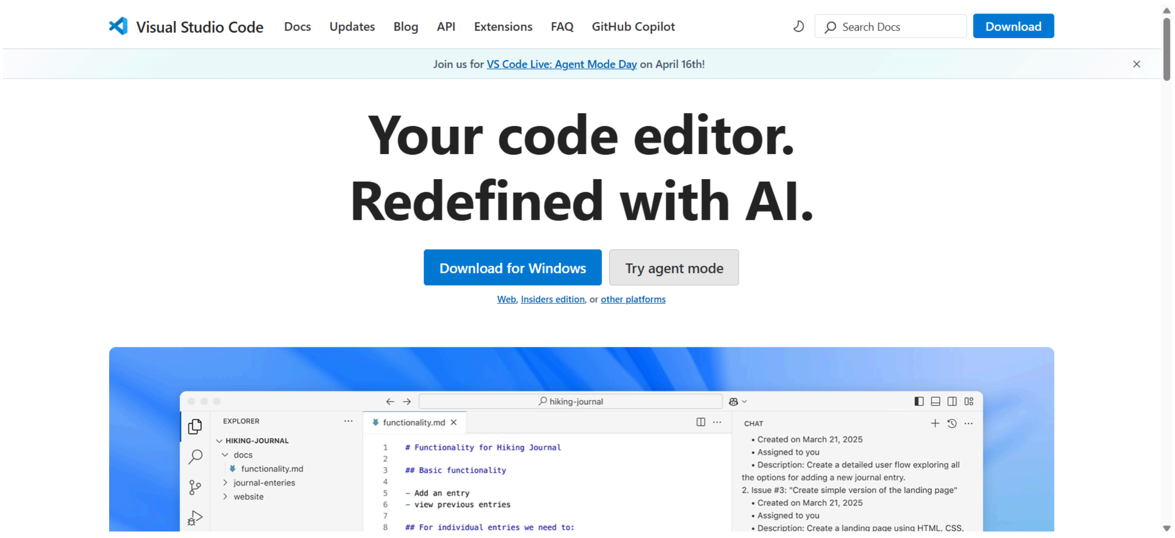
Install Required Tools

Step 7: Install Visual Studio Code and Android SDK

- If flutter doctor indicates that the Android SDK or a compatible IDE is missing, you'll need to install Visual Studio Code (VS Code) and configure the Android SDK manually. Follow these steps:

1) **Download Visual Studio Code**

- Go to the official VS Code website (https://code.visualstudio.com/) and download the latest installer for Windows.


Visual Studio Code
Docs
Updates
Blog
API
Extensions
FAQ
GitHub Copilot
Search Docs
Download
Join us for VS Code Live: Agent Mode Day on April 16th!
Your code editor. Redefined with AI.
Download for Windows
Try agent mode
Web, Insiders edition, or other platforms

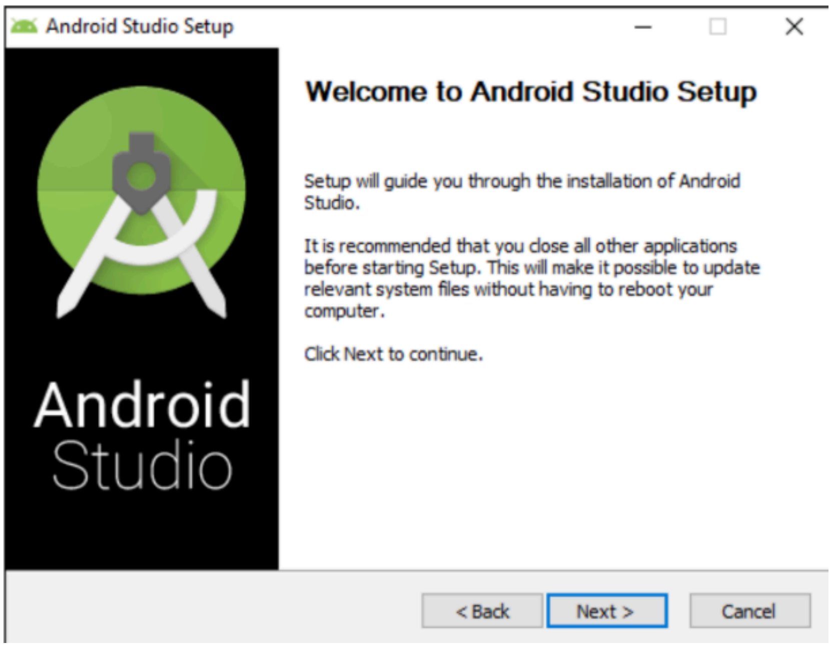
2) **Install VS Code**

- Once downloaded, run the .exe file. Follow the installation wizard to complete the setup.

3) **Install Flutter and Dart Extensions**

- Open VS Code, go to the Extensions view (Ctrl+Shift+X), and search for Flutter and Dart. Install both extensions to enable Flutter development support in VS Code.

4) **Install the Android SDK**


Android Studio Setup

Welcome to Android Studio Setup

Setup will guide you through the installation of Android Studio.

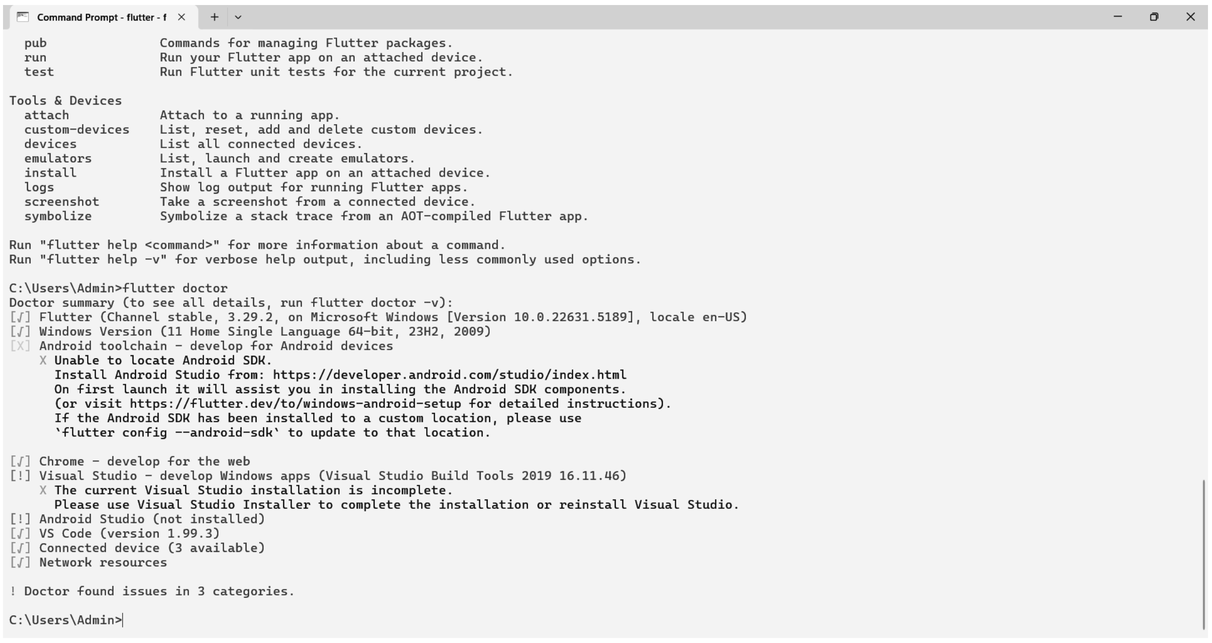
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

.

5) **Accept Android Licenses**

- Run the following commands in the Command Prompt:
  flutter doctor
  flutter doctor --android-licenses
  Follow the prompts to accept all Android SDK licenses.



```
  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.

C:\Users\Admin>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.29.2, on Microsoft Windows [Version 10.0.22631.5189], locale en-US)
[√] Windows Version (11 Home Single Language 64-bit, 23H2, 2009)
[X] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/to/windows-android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.

[√] Chrome - develop for the web
[!] Visual Studio - develop Windows apps (Visual Studio Build Tools 2019 16.11.46)
    X The current Visual Studio installation is incomplete.
      Please use Visual Studio Installer to complete the installation or reinstall Visual Studio.
[!] Android Studio (not installed)
[√] VS Code (version 1.99.3)
[√] Connected device (3 available)
[√] Network resources

! Doctor found issues in 3 categories.

C:\Users\Admin>
```

Step 8: Enable Flutter Web Support and Run on Chrome

- To test your Flutter app in a web browser like Chrome, you need to enable Flutter's web support and configure VS Code to run the app in Chrome. Follow these steps:

Name:Bhushan Mukund Kor **Academic Year:2024-2025**

Division: D15C **Roll No: 28**

1) Enable Web Support in Flutter

- Ensure your Flutter SDK is up to date and supports web development. Open a Command Prompt and run:

```
flutter channel stable
flutter upgrade
```

- This ensures you're on the stable channel with the latest Flutter version.
- Enable web support by running:

```
flutter config --enable-web
```

This command enables the web renderer for Flutter, allowing you to target Chrome as a runtime environment.

2) Create or Open a Flutter Project

- If you don't already have a Flutter project, create one by running:

```
flutter create my_app
cd my_app
```

- Replace my_app with your desired project name.
- Open the project in VS Code:
- Launch VS Code.
- Go to File > Open Folder and select the project folder (e.g., my_app).

3) Configure VS Code to Run on Chrome

- Ensure the Flutter and Dart extensions are installed in VS Code (as described in the previous response, Step 7.3).
- Open the pubspec.yaml file in your project and ensure it includes the Flutter web dependencies. Most Flutter projects created after web support became stable (Flutter 2.0+) include these by default. If not, you don't need to modify anything manually unless you're using specific web-related packages.
- In VS Code, go to the Run and Debug panel (Ctrl+Shift+D).
- Click create a launch.json file (if it doesn't exist) and select Dart & Flutter as the environment.
- Modify the launch.json file in the .vscode folder to include a configuration for web, like this:

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Run on Chrome",
      "type": "dart",
      "request": "launch",
      "program": "lib/main.dart",
      "deviceId": "web-server",
      "args": ["--web-renderer", "auto"]
    }
    ]
  }
```

This sets up VS Code to run your app on Chrome via Flutter's web server.

4) Run the App on Chrome

In VS Code, go to the Device Selector (bottom-right corner) and select Chrome (web-javascript) or Web Server from the list of available devices. If Chrome doesn't appear, ensure web support is enabled (Step 8.1) and Chrome is installed.
Open the terminal in VS Code (`Ctrl+``) and run:
bash

flutter run
Alternatively, press F5 or click Run > Start Debugging with the "Run on Chrome" configuration selected in the Run and Debug panel.
Flutter will build the app for the web and launch Chrome automatically, displaying your app at a URL like http://localhost:port (e.g., http://localhost:53599).

Conclusion:
The successful installation and configuration of the Flutter environment, as outlined in this experiment, enables developers to create and test cross-platform applications efficiently. By setting up the Flutter SDK, integrating Visual Studio Code with Flutter and Dart extensions, configuring the Android SDK, and enabling web support to run the app on Chrome, a robust development environment is established. This setup eliminates the need for physical devices or emulators during initial testing, streamlining the development process and allowing for rapid iteration and debugging of Flutter applications directly in a web browser.