Aim: Installation and Configuration of Flutter Environment.

Theory:

Flutter is an open-source UI toolkit developed by Google that enables the development of applications for multiple platforms including Android, iOS, web, desktop, and embedded systems using a single codebase. It uses the Dart programming language and offers a rich set of pre-designed widgets for crafting beautiful and natively compiled applications.

Key Features of Flutter:

- Single Codebase: Write once and run on multiple platforms.
- Hot Reload: Instantly see changes during development.
- Custom Widgets: Access to a wide collection of widgets and customization options.
- High Performance: Compiles to ARM or x86 native libraries.
- Open Source: Free to use and backed by an active community.

Requirements

For Windows:

- Windows 10 or later (64-bit)
- 1.64 GB of disk space (excluding IDE/tools)
- Git for Windows

For macOS:

- macOS (Intel or Apple Silicon)
- Xcode for iOS development

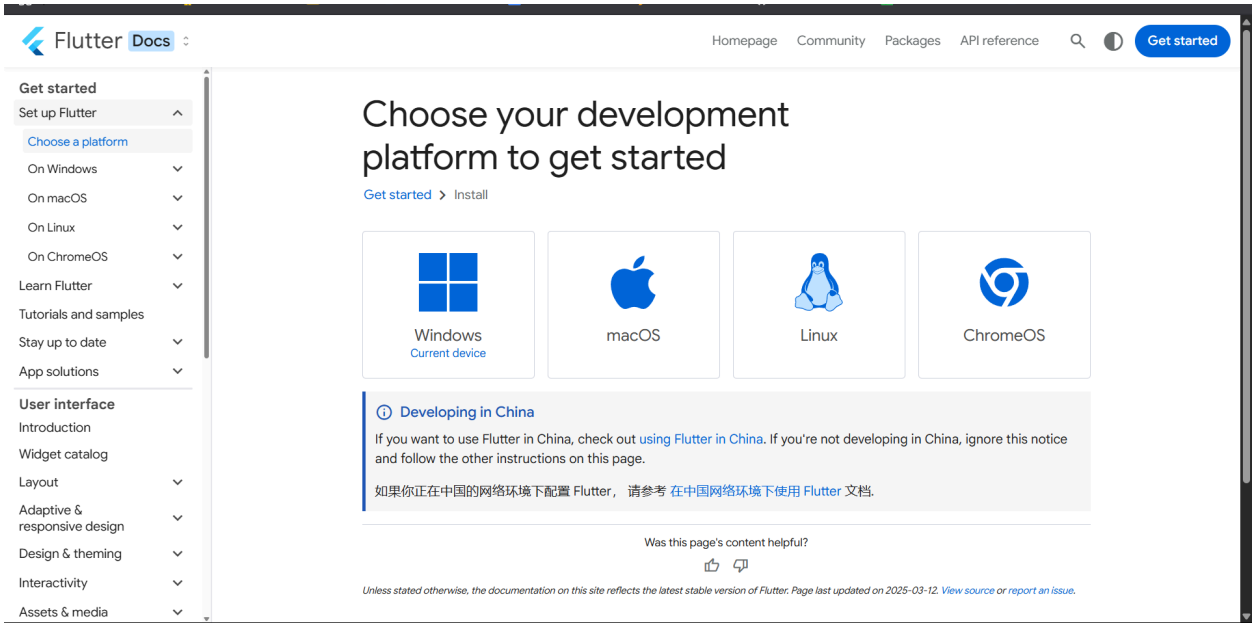
For Linux:

- Any recent Linux distribution with bash, curl, git, and unzip

Installation Steps

Step 1: Download the Flutter SDK

- Navigate to the official Flutter website: https://flutter.dev
- Select your operating system and download the latest stable release.



- Extract the contents to a desired location (e.g., C:\flutter on Windows or ~/development/flutter on Linux/macOS).

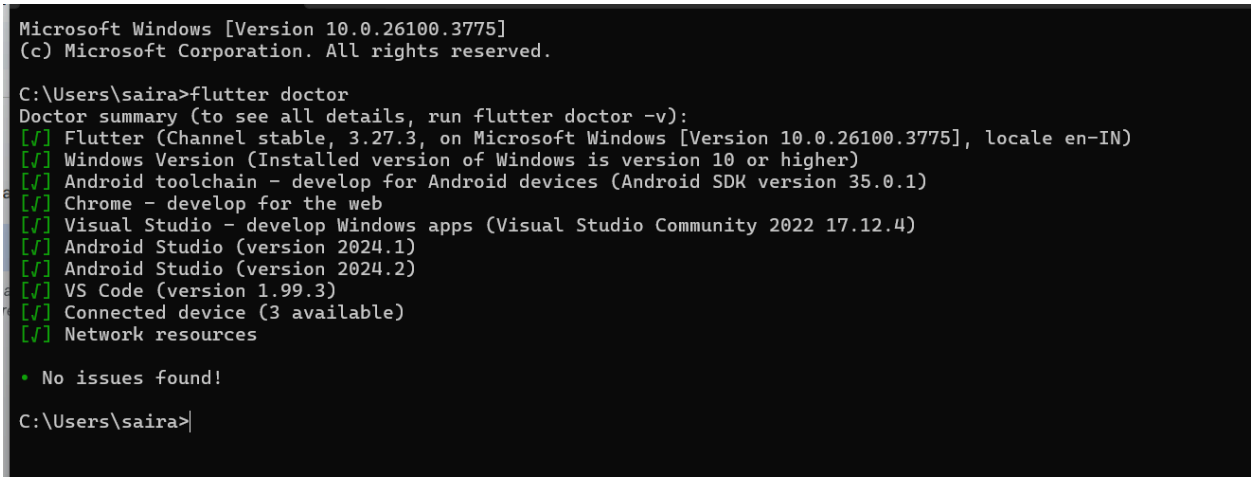
Step 2: Set Up System Environment Variable

- Add the Flutter SDK's bin directory to your system PATH.
    - On Windows: Add C:\flutter\bin to Environment Variables.
    - On macOS/Linux: Add the line export PATH="$PATH:[PATH_TO_FLUTTER_DIRECTORY]/bin" to your shell configuration file (e.g., .bashrc, .zshrc).

Step 3: Verify Installation

Open the terminal or command prompt and run:

flutter doctor



This command checks your environment and displays a report of the Flutter installation along with required dependencies.

Step 4: Install a Code Editor or IDE

You can use:

- Android Studio (recommended for full-featured Flutter development)
- Visual Studio Code (lightweight and fast)

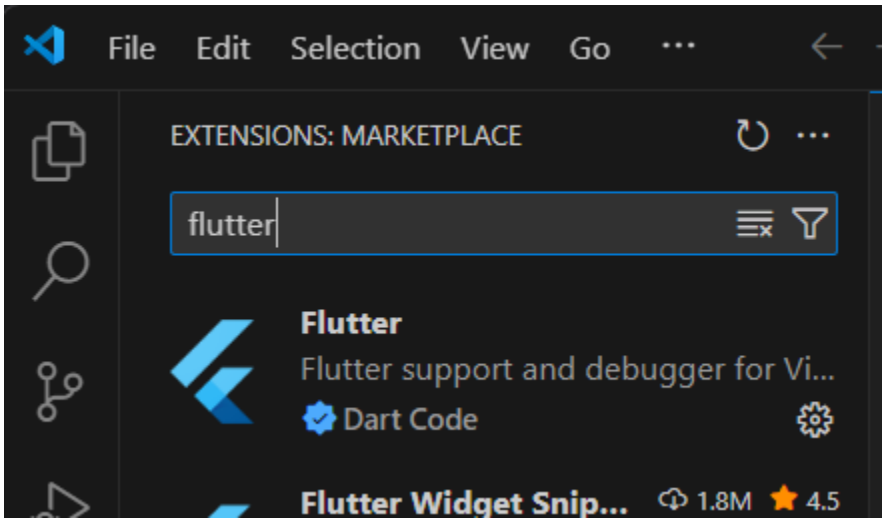
Step 5: Install Flutter and Dart Plugins

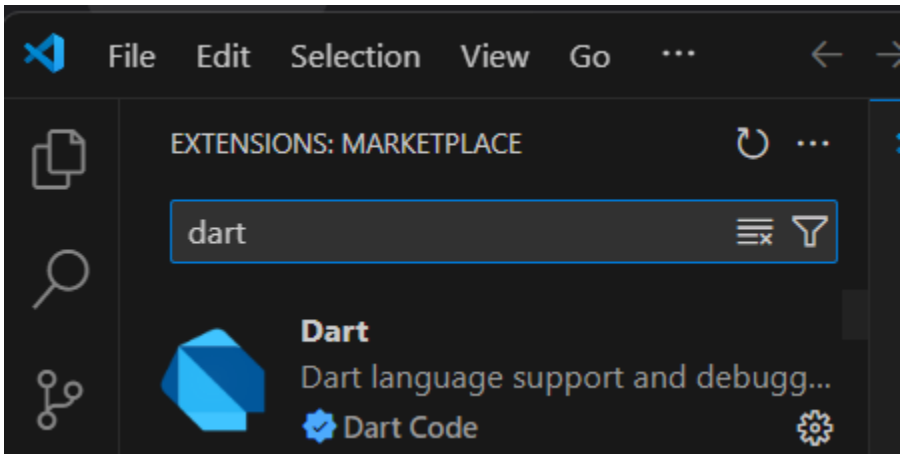
In Android Studio:

- Open Settings > Plugins
- Search and install the Flutter plugin
- The Dart plugin will be prompted for installation automatically

In Visual Studio Code:

- Open the Extensions tab
- Search for and install both "Flutter" and "Dart"





Step 6: Set Up an Android Emulator (Optional for Testing)

- Open Android Studio
- Go to AVD Manager and create a new virtual device
- Choose a hardware profile and system image
- Launch the emulator

Step 7: Create and Run a Sample Application

Use the following commands to create a basic Flutter project and run it:

```
flutter create myapp
cd myapp
flutter run
```

Make sure a simulator or physical device is connected and running.

Conclusion: The Flutter environment has been successfully installed and configured. Developers can now build and test cross-platform applications using a single codebase. The setup ensures a consistent development experience with native performance and modern UI capabilities.