

Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103

Department of Information Technology

A.Y. 2021-2022

Class: TE-ITA/B, Semester: VI

Subject: **MAD & PWA LAB**

Experiment – 1A: Installation and Configuration of Flutter Environment.

1. **Aim:** To install and configure Flutter Environment.
2. **Objectives:** After study of this experiment, the student will be able to
 - Learn the basics of the Flutter framework.
 - Install and configure Flutter Environment.
 - Execute program in flutter environment.
3. **Outcomes:** After study of this experiment, the student will be able to
 - Understand cross platform mobile application development using Flutter framework.
(L604.1)
4. **Prerequisite:** None.
5. **Requirements:** Personal Computer, 8 GB RAM, Internet Connection, Web browser.

6. Pre-Experiment Exercise:

Brief Theory:

Flutter framework

Flutter is a UI toolkit for building fast, beautiful, natively compiled applications for mobile, web, and desktop with one programming language and single codebase. It is free and open source. Initially, it was developed from Google and now managed by an ECMA standard. Flutter apps use Dart programming language for creating an app.

Dart

Dart is an open-source general-purpose programming language. It is originally developed by Google and later approved as a standard by ECMA. It is an object-oriented programming language that bears resemblance to C, Java and Javascript.

Dart language is widely used to develop Android applications, iOS applications, IoT applications and web applications, using the Flutter framework. It supports application development in both client side and server side. A popular example of Dart application is Gmail.

7. Laboratory Exercise

A. Procedure

1. List down the steps to install Flutter framework.

Step 1: Download the installation bundle of the Flutter Software Development Kit for windows. To download Flutter SDK, go to its official website, click on Get started button

Step 2: Next, to download the latest Flutter SDK, click on the Windows icon.

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

Step 4. Open environment variable and add flutter /bin in path section.

Step 5. Open cmd and run flutter doctor. This command checks your environment and displays a report of the status of the flutter application check the output for other software's you might require to install.

Step 6. Install android studio to get the android SDK. Once download open android studio and install and latest android SDK, command line tools and android SDK build tools.

Step 7. Run \$ flutter doctor –android licenses and sign all the licenses.

Step 8. Use android studio, visual studio or any preferred IDE to code you flutter applications.

B. Result/Observation

1. Installation screenshots.
2. Print out of program code and output.

The screenshot shows the Flutter documentation website at docs.flutter.dev/get-started/install/windows. The page title is "Windows install". The navigation bar includes links for Multi-Platform, Development, Ecosystem, Showcase, Docs, and a "Get started" button. A sidebar on the left provides links for "Get started" (1. Install, 2. Set up an editor, 3. Test drive, 4. Write your first app, 5. Learn more), "From another platform?", "Samples & tutorials", and "Get the Flutter SDK". The main content area features a "System requirements" section with a bulleted list: Operating Systems (Windows 7 SP1 or later (64-bit), x86-64 based), Disk Space (1.64 GB (does not include disk space for IDE/tools)), and Tools (Flutter depends on these tools being available in your environment: Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10) and Git for Windows 2.x, with the Use Git from the Windows Command Prompt option). It also notes that if Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell. Below this is a "Get the Flutter SDK" section.

The screenshot shows the Flutter documentation website at docs.flutter.dev/get-started/install/windows. The main content is titled "Get the Flutter SDK". It provides instructions for downloading the Flutter SDK, extracting the zip file, and running commands in the Flutter Console. A warning box advises against installing Flutter in a directory requiring elevated privileges. The sidebar contains links for "Get started", "Samples & tutorials", and "Development". The top navigation bar includes "Multi-Platform", "Development", "Ecosystem", "Showcase", "Docs", and "Get started".

Get the Flutter SDK

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

[flutter_windows_2.10.1-stable.zip](#)

For other release channels, and older builds, see the [SDK releases](#) page.

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\Users\<your-user-name>\Documents`).

⚠ Warning: Do not install Flutter in a directory like `C:\Program Files` that requires elevated privileges.

If you don't want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the [Flutter repo](#) on GitHub, and change branches or tags as needed. For example:

```
C:\src>git clone https://github.com/flutter/flutter.git -b stable
```

You are now ready to run Flutter commands in the Flutter Console.

Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the `PATH` environment variable:

The screenshot shows the Android Studio download page at developer.android.com/studio. The page features the Android Studio logo and a brief description of its capabilities. It includes a "Download" button, "What's new", "User guide", and "Preview" sections. Below the main content are "Download options" and "Release notes" buttons. The top navigation bar includes links for "developers", "Platform", "Android Studio" (which is underlined), "Google Play", "Jetpack", "Kotlin", "Docs", and "Games". There is also a search bar and a sign-in link.

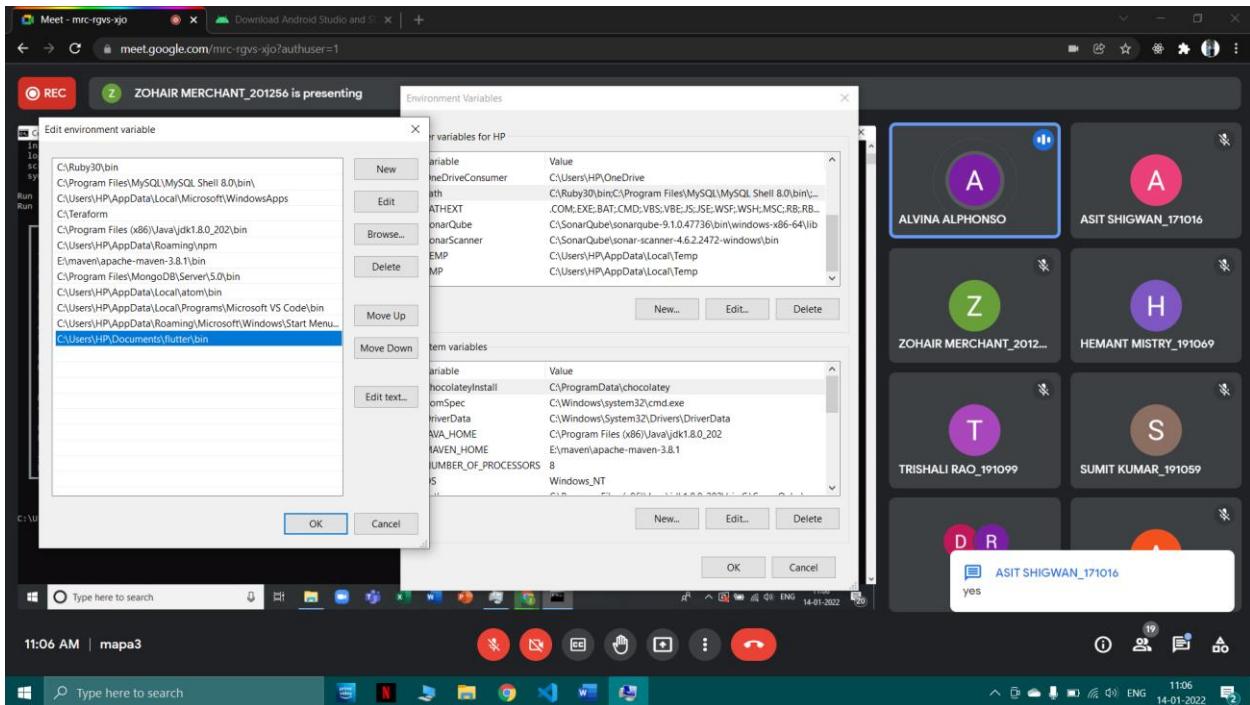
android studio

Android Studio provides the fastest tools for building apps on every type of Android device.

[Download Android Studio](#)

Android Studio Bumblebee | 2021.1.1 Patch 1 for Windows 64-bit (872 MiB)

[Download options](#) [Release notes](#)



```

Windows PowerShell
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.
format       Format code on every Dart file.
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 or create emulators.
install      Install Flutter on an attached device.
log          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.

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage statistics and basic crash reports. This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, type 'Flutter config --no-analytics'. To display the current setting, type 'Flutter config'. If you opt out of analytics, an opt-out event will be sent, and then no further information will be sent by the Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

```

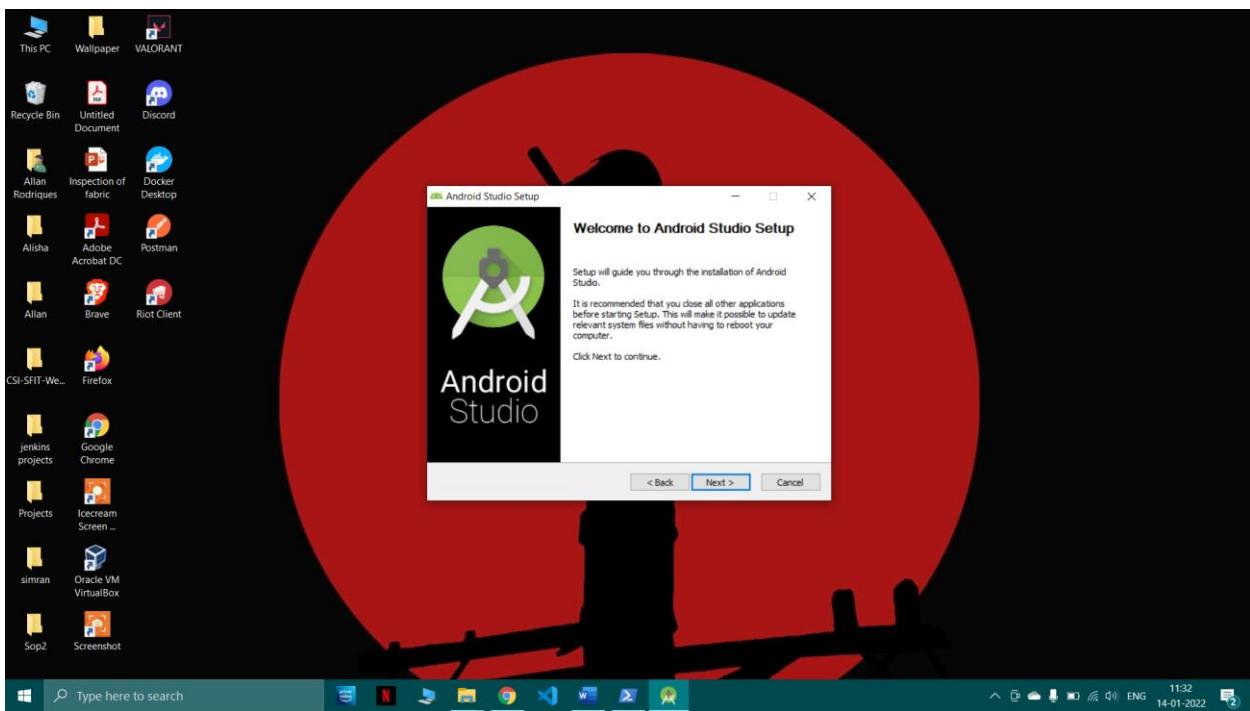
```
Windows PowerShell

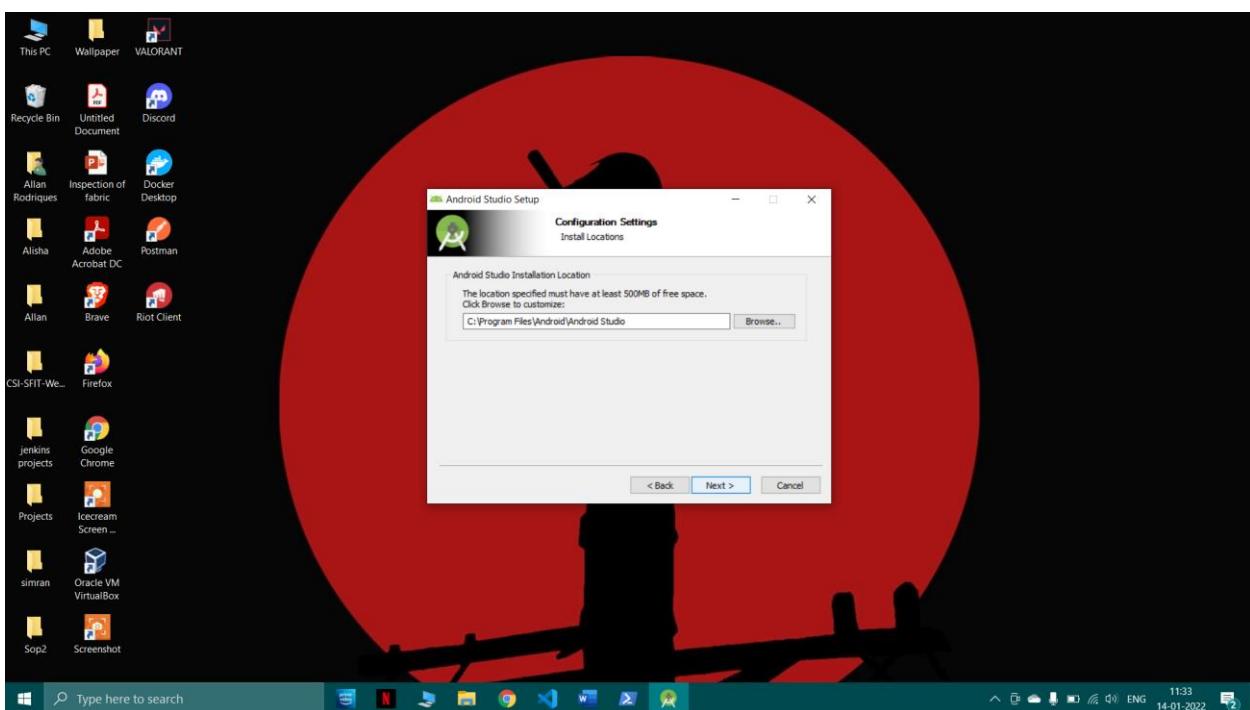
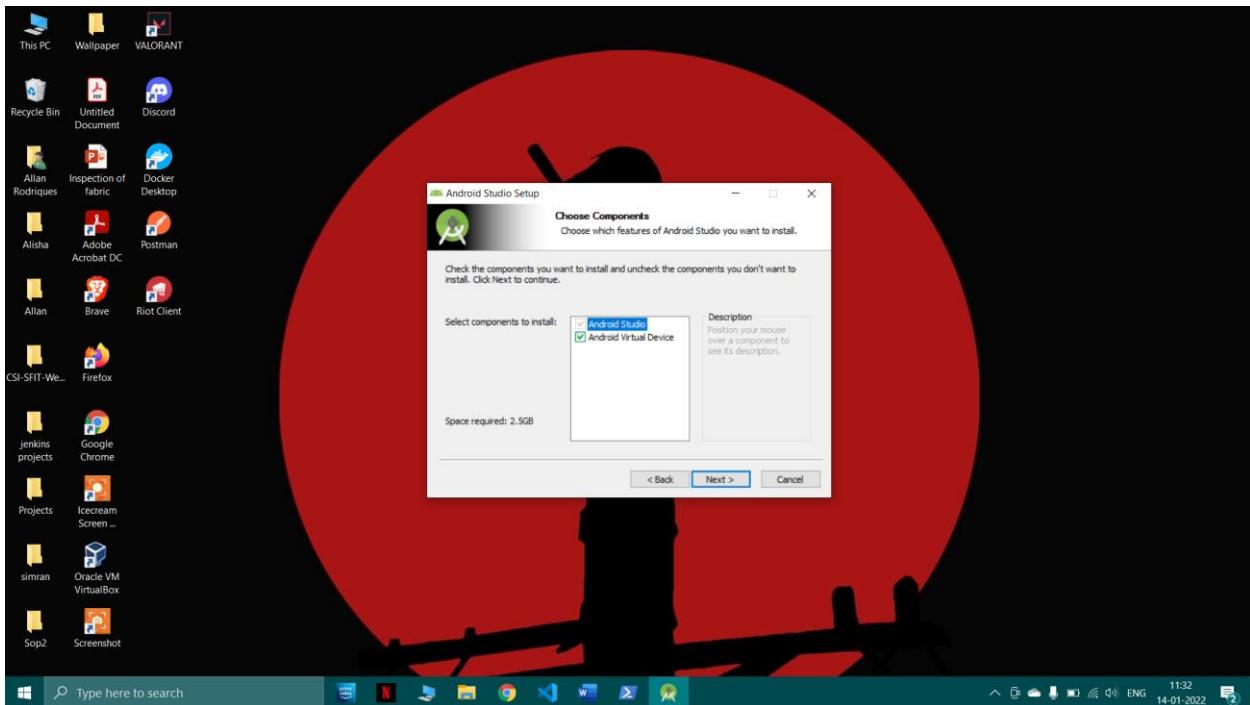
PS C:\Users\HP> flutter doctor
Running "flutter doctor --android-tools"...
Doctor summary (to see all details, run flutter doctor -v):
  Doctor found no issues in 1 category.

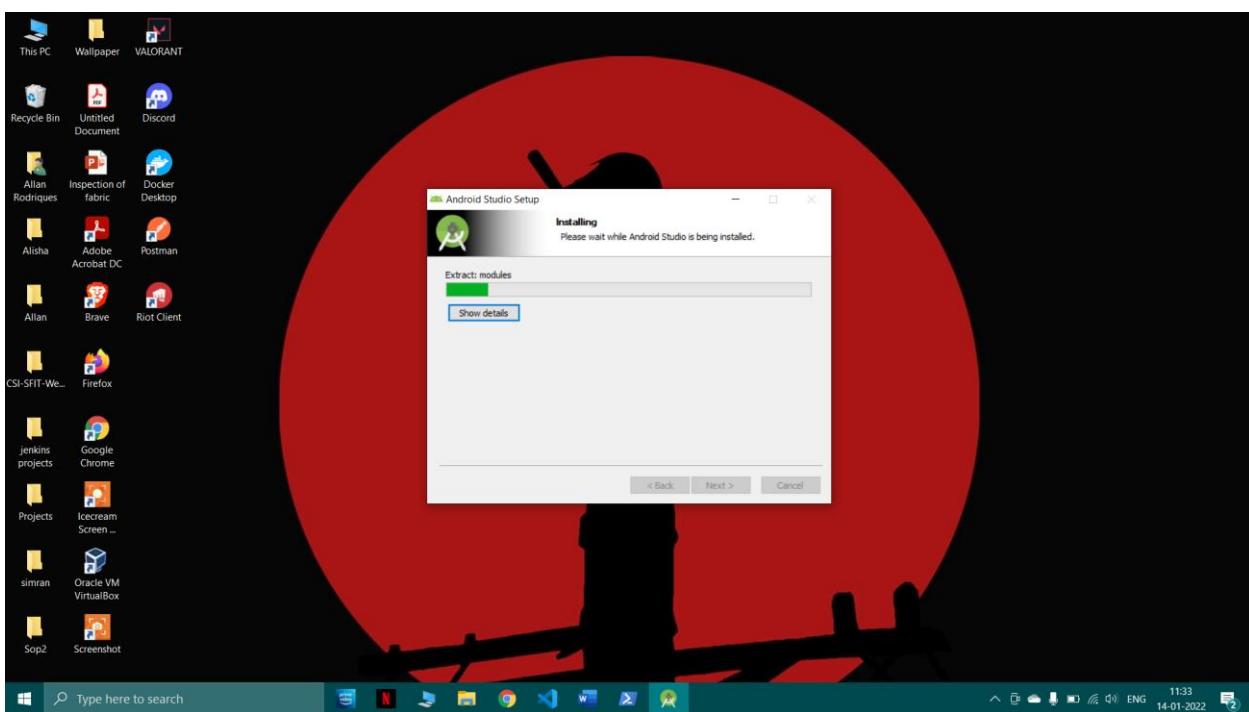
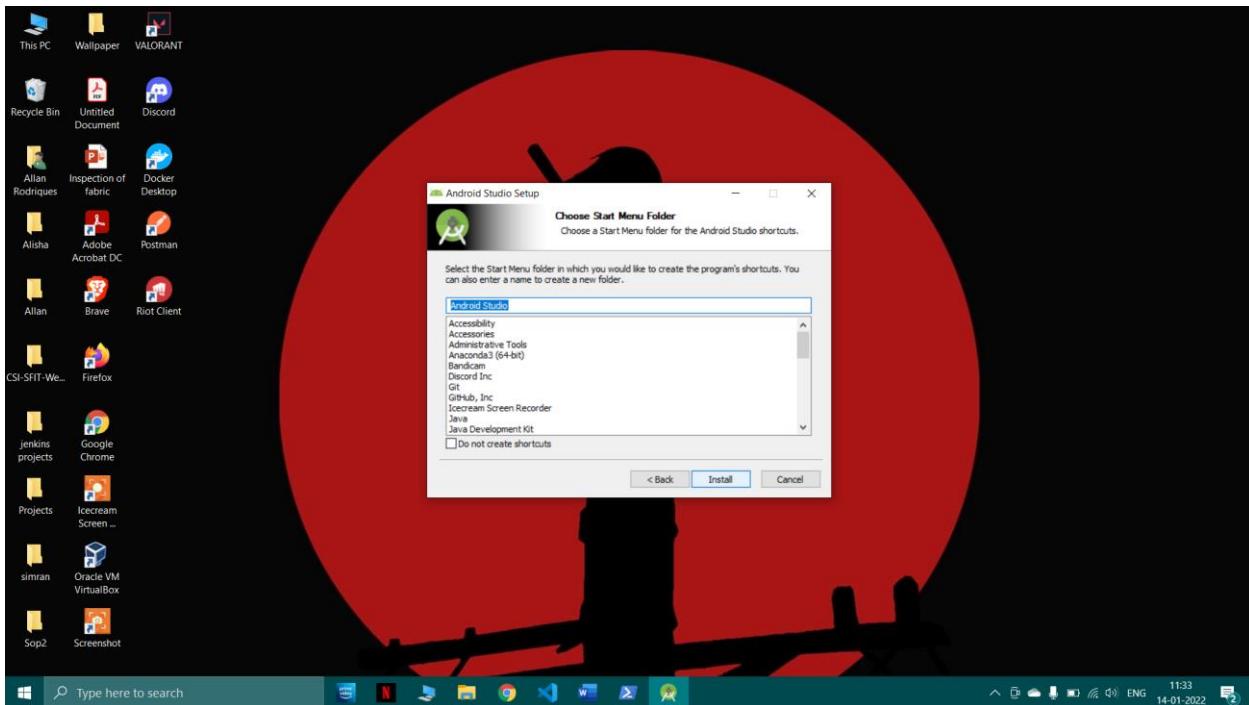
[!] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from https://developer.android.com/studio/index.html
      Or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instructions.
        If the Android SDK has been installed to a custom location, please use
          `flutter config --android-sdk` to update to that location.

[!] Chrome - deployer for the web
    [!] Android Studio (not installed)
    VS Code (version 1.61.2)
    Connected device (2 available)

1 doctor found issue in 2 categories.
PS C:\Users\HP>
```







File Edit Selection View Go Run Terminal Help

main.dart X

VARIABLES

```
lib > main.dart > MyApp
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: key);
9
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // Try running your application with "flutter run". You'll see
19       // application has a blue toolbar. Then, without quitting the
20       // changing the primarySwatch below to Colors.green and then i
21       // "hot reload" (press "r" in the console where you ran "flutt
22       // or simply save your changes to "hot reload" in a Flutter ID
```

WATCH

CALL STACK RUNNING

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

"Install Android SDK Platform 31 (revision: 1)" complete.
"Install Android SDK Platform 31 (revision: 1)" finished.
Built build\app\outputs\flutter-apk\app-debug.apk.
Connecting to VM Service at ws://127.0.0.1:53675/DxFv7L-v9sI=/ws

Source: Dart (Extens... Open Always Open Not Now Never Ask

12:28 14-01-2022

File Edit Selection View Go Run Terminal Help

main.dart X

VARIABLES

```
lib > main.dart > MyApp
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: key);
9
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // Try running your application with "flutter run". You'll see
19       // application has a blue toolbar. Then, without quitting the
20       // changing the primarySwatch below to Colors.green and then i
21       // "hot reload" (press "r" in the console where you ran "flutt
22       // or simply save your changes to "hot reload" in a Flutter ID
```

WATCH

CALL STACK

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

Launching lib\main.dart on Android SDK built for x86 in debug mode...

Running Gradle task 'assembleDebug'...

12:17 14-01-2022

The screenshot shows two instances of Visual Studio Code side-by-side. Both instances have the same project structure and code editor open, displaying the file `main.dart`. The code defines a simple Flutter application with a `MaterialApp` widget.

```
lib > main.dart > MyApp
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({Key? key}) : super(key: key);
9
10 // This widget is the root of your application.
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     title: 'Flutter Demo',
15     theme: ThemeData(
16       // This is the theme of your application.
17       //
18       // Try running your application with "flutter run". You'll see
19       // application has a blue toolbar. Then, without quitting the
20       // changing the primarySwatch below to Colors.green and then i
21       // "hot reload" (press "r" in the console where you ran "flutt
22       // or simply save your changes to "hot reload" in a Flutter ID
23       // ...
24   );
25 }
```

The top instance shows an `Android` virtual device running the application. The bottom instance shows the `Android Virtual Device Manager` open, listing a single device entry:

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Pixel 2 XL API 23	Pixel 2 XL API 23		1440 x 2880	23	Android 6.0 (Google Play)	x86	2.5 GB	▶️ 📁 ⚙️

Both instances also show the terminal output indicating a successful build and run:

```
Your application code is in .\lib\main.dart.
exit code 0
```

The status bar at the bottom of both VS Code windows shows the following information:

- Ln 1, Col 1 / Spaces: 2 / UTF-8 / Dart / Dart DevTools / Flutter: 2.8.1
- Pixel 2 XL API 23 (android-x86 emulator) / 12:16 / ENG / 14-01-2022

Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

AVD Name: Pixel 2 XL API 23

Pixel 2 XL 5.99 1440x2880 560dpi Change...

Marshmallow Android 6.0 x86 Change...

Startup orientation: Portrait

Emulated Performance: Graphics: Automatic

Device Frame: Enable Device Frame

Show Advanced Settings

AVD Name

The name of this AVD.

Previous Next Cancel Finish

System Image

Select a system image

Recommended x86 Images Other Images

Release Name	API Level	ABIs	Target
R Download	30	x86	Android 11.0 (Google APIs)
Q Download	29	x86	Android 10.0 (Google APIs)
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Neuter Download	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Marshmallow	23	x86	Android 6.0 (Google APIs)
Lollipop Download	22	x86	Android 5.1 (Google APIs)

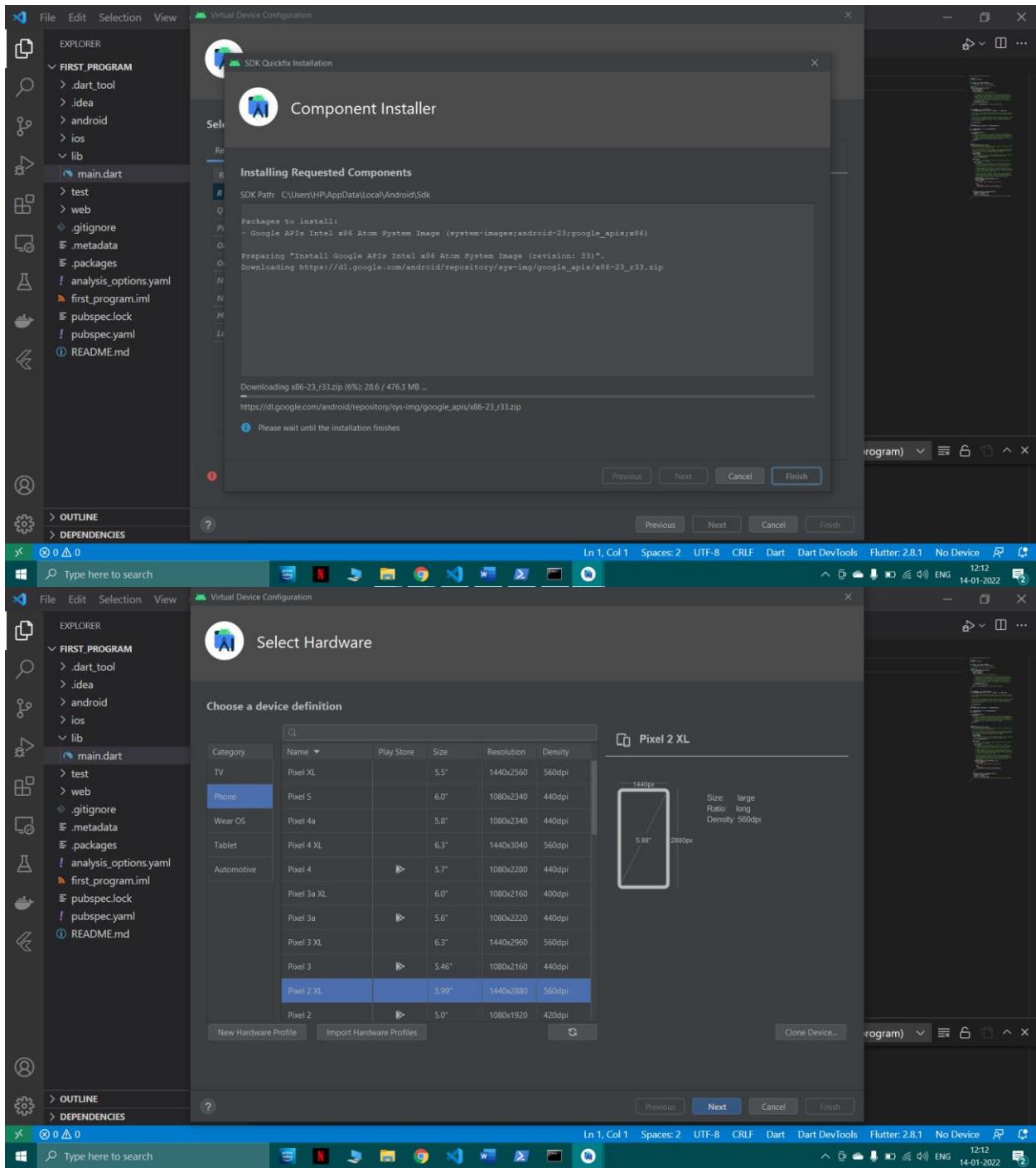
Marshmallow

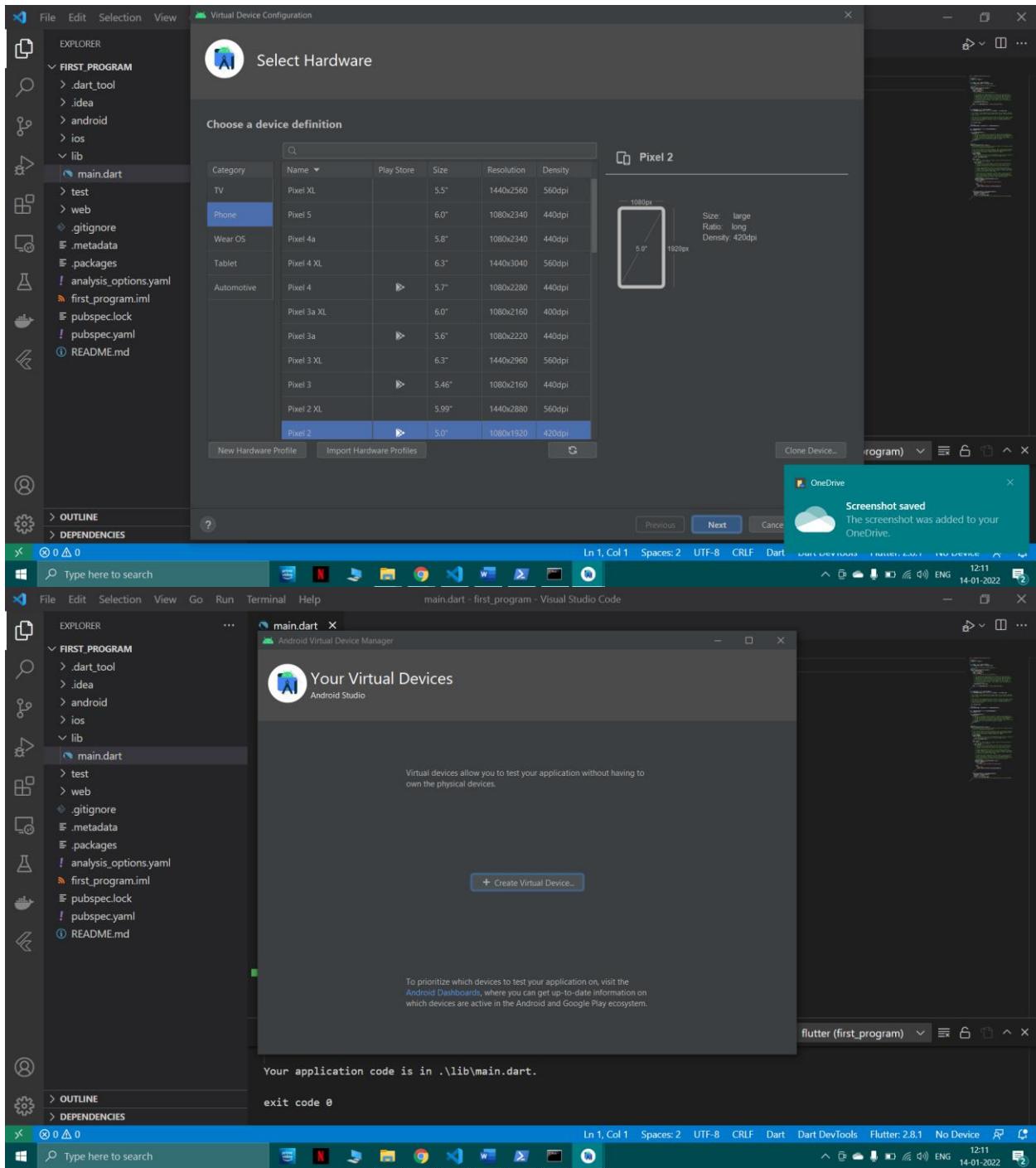
API Level: 23
Android: 6.0
Google Inc.
System Image
x86

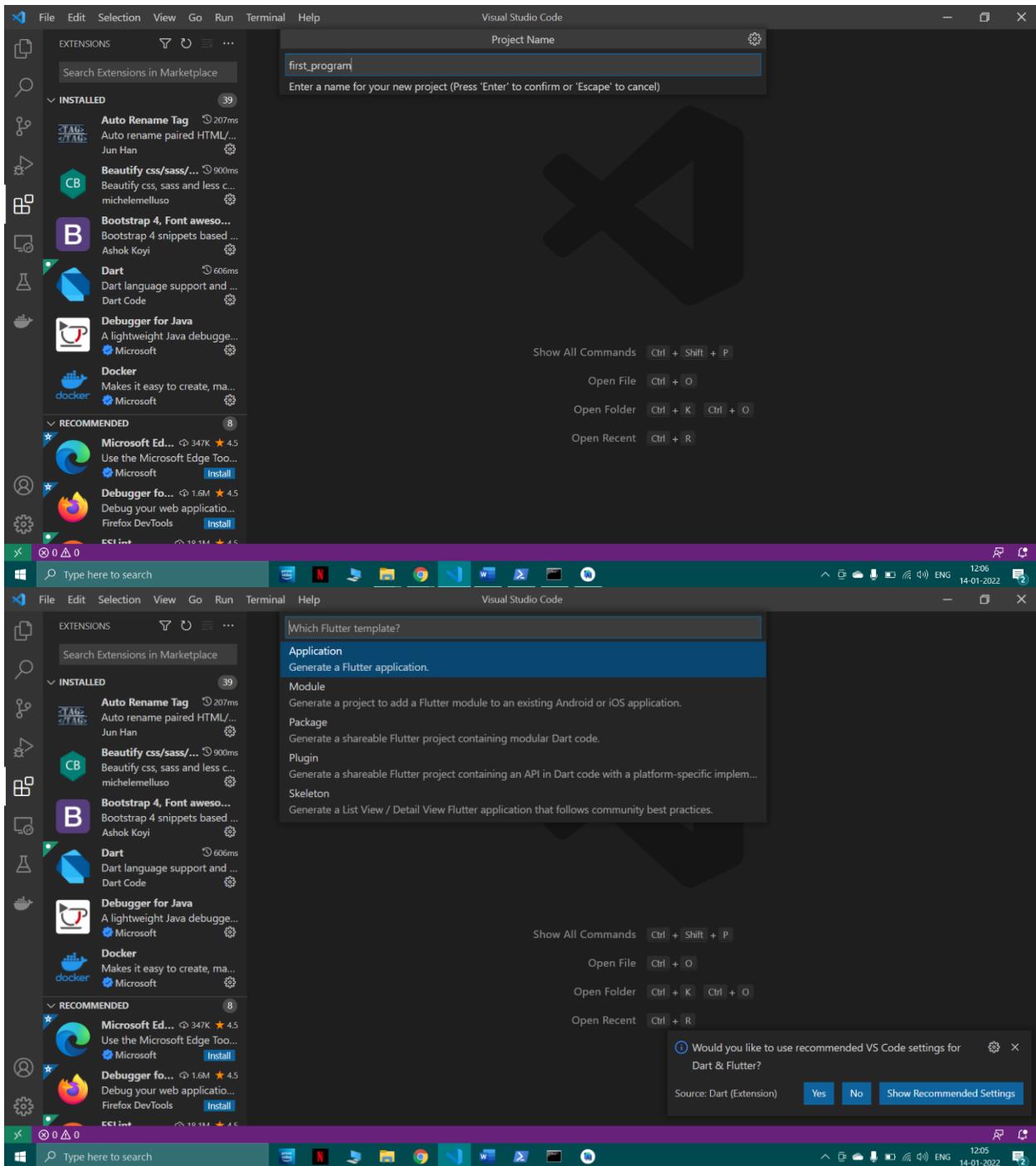
We recommend these images because they run the fastest and support Google APIs.

Questions on API level?
See the [API level distribution chart](#)

Previous Next Cancel Finish







Extension: Dart - Visual Studio Code

Dart v3.32.0

Dart Code | 4,006,346 | ★★★★★ (54)

Dart language support and debugger for Visual Studio Code.

[Disable](#) [Uninstall](#) [Switch to Pre-Release Version](#)

This extension is enabled globally.

Details **Feature Contributions** **Changelog** **Runtime Status**

[chat](#) [discord](#) [chat](#) [gitter](#) [twitter](#) [dartcode](#) [help](#) [contribute](#)

Introduction

Dart Code extends VS Code with support for the Dart programming language, and provides tools for effectively editing, refactoring, running, and reloading Flutter mobile apps, and AngularDart web apps.

Installation

Dart Code can be [installed from the Visual Studio Code Marketplace](#) or by [searching within VS Code](#).

Features

- Edit and Debug Flutter mobile apps (launch using F5 or the Debug menu)
- Edit and Debug Dart command line apps (launch using F5 or the Debug menu)
- Automatic hot reloads for Flutter
- Refactorings and Code fixes (lightbulb)
- Quickly switch between devices for Flutter

Extension: Flutter - Visual Studio Code

Flutter v3.32.0

Dart Code | 3,725,300 | ★★★★★ (56)

Flutter support and debugger for Visual Studio Code.

[Disable](#) [Uninstall](#)

This extension is enabled globally.

Details **Feature Contributions** **Dependencies** **Runtime Status**

Introduction

This VS Code extension adds support for effectively editing, refactoring, running, and reloading Flutter mobile apps, as well as support for the Dart programming language.

More Info

Released on 4/18/2018, 22:19:03
Last updated 1/2/2022, 17:56:52
Identifier dart-code.flutter

Categories

Programming Languages
Snippets
Linters
Formatters
Debuggers

Resources

Marketplace
Repository
License

Code Editor Screenshot

main.dart

```

void _incrementCounter() {
  setState(() {
    _counter++;
  });
}

@Override
Widget build(BuildContext context) {
  return new Scaffold(
    appBar: new AppBar(
      title: new Text(widget.title),
    ),
    body: new Center(
      child: new Text(
        'Button clicked $_counter times',
        style: Theme.of(context).textTheme.display1,
      ),
    ),
  );
}

```

Flutter Demo Home Page

Button clicked 0 times.

```
Command Prompt
le law or applicable court decisions.

10.6 Export Regulations / Export Control. Recipient shall not export, either directly or indirectly, any product, service or technical data or system incorporating the Evaluation Materials without first obtaining any required license or other necessary approval from the U.S. Department of Commerce or any other governing agency or department of the United States Government. In the event any product is exported from the United States or re-exported from a foreign destination by Recipient, Recipient shall ensure that the distribution and export/re-export or import of the product is in compliance with all applicable laws, regulations, orders, or other restrictions of the U.S. Export Administration Regulations and the appropriate foreign government. Recipient agrees that neither it nor any of its subsidiaries will export/re-export any technical data, process, product, or service, directly or indirectly, to any country for which the United States government or any agency thereof or the foreign government from where it is shipping requires an export license, or other governmental approval, without first obtaining such license or approval. Recipient also agrees to implement measures to ensure that foreign national employees are authorized to receive any information controlled by U.S. export control laws. An export is "deemed" to take place when information is released to a foreign national wherever located.

10.7 Special Terms for Pre-Release Materials. If so indicated in the description of the Evaluation Software, the Evaluation Software may contain Pre-Release Materials. Recipient 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) MIPS can provide no assurance that it will ever produce or make generally available a production version of the Pre-Release Materials. MIPS 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 Recipient or any other person.

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

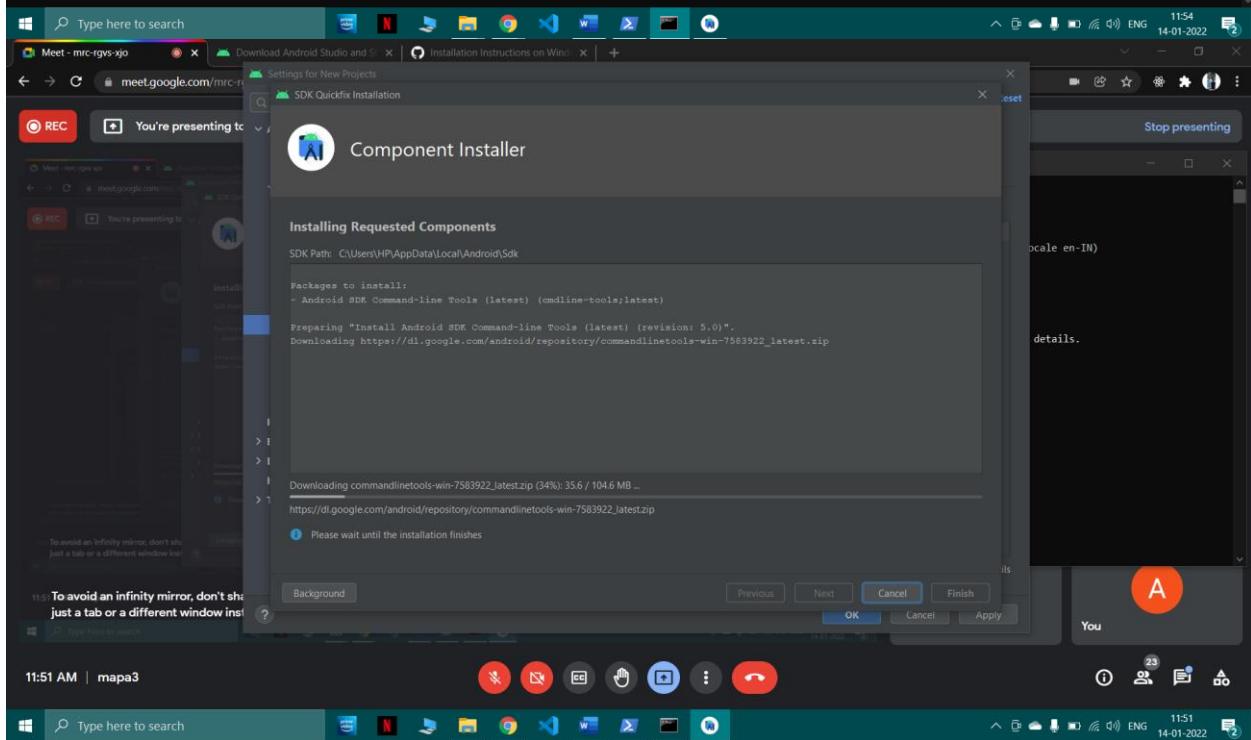
10.8 Open Source Software. In the event Open Source software is included with Evaluation Software, such Open Source software is licensed pursuant to the applicable Open Source software license agreement identified in the Open Source software comments in the applicable source code file(s) and/or file 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.

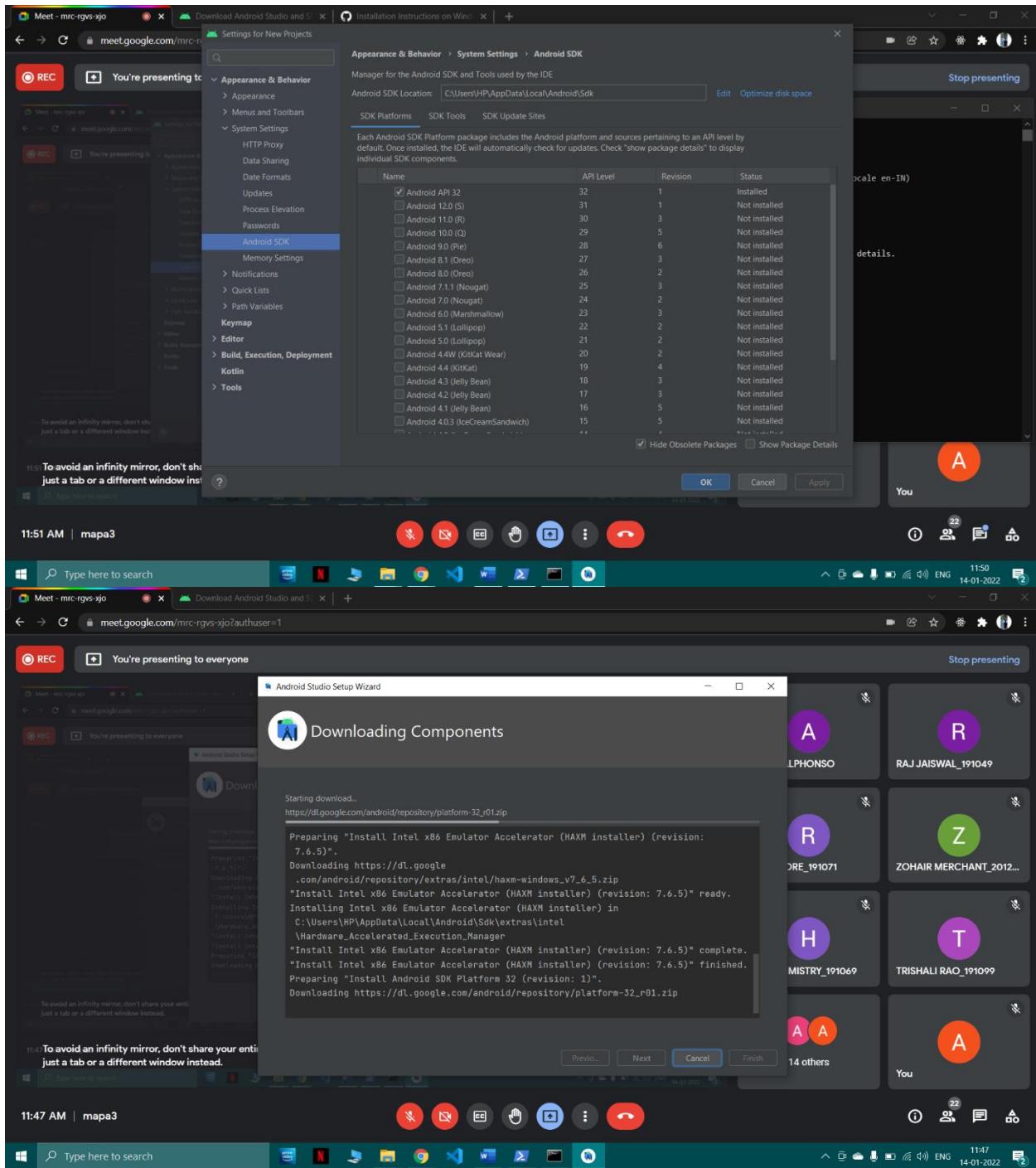
Accept? (y/n): y
All SDK package licenses accepted

C:\Users\HP>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0.19043.1415], locale en-IN)
[✓] Android toolchain - develop for Android devices (Android SDK version 32.0.0)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2020.3)
[✓] VS Code (version 1.63.2)
[✓] Connected device (2 available)

• No issues found!

C:\Users\HP>
```





```

Windows PowerShell
PS C:\Users\HP> flutter doctor
Running flutter doctor -v...                                17.6s
Doctor summary (to see all details, run flutter doctor -v):
  [!] Android toolchain - develop for Android devices
      X Unable to locate Android SDK.
        Install Android Studio from: https://developer.android.com/studio/index.html
        Or download from: https://storage.googleapis.com/android-sdk/index.html (for API levels 14+)
        (or visit https://flutter.dev/docs/get-started/install/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
  [!] Android Studio (not installed)
  VS Code (version 1.63.2)
  Connected device (2 available)

! Doctor found issues in 2 categories.
PS C:\Users\HP>

```



```

Windows PowerShell
Type here to search
Project
  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.
    dev       Run instrumentation tests on the project on an attached device or emulator.
    format    Format one or more Dart files.
    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 all available 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 <commands>" for more information about a command.
Run "Flutter help -v" for verbose help output, including less commonly used options.

Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage statistics and basic crash reports. This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, type "flutter config --no-analytics". To display the current setting, type "flutter config". If you opt out of analytics, an opt-out event will be sent, and then no further information will be sent by the Flutter tool.

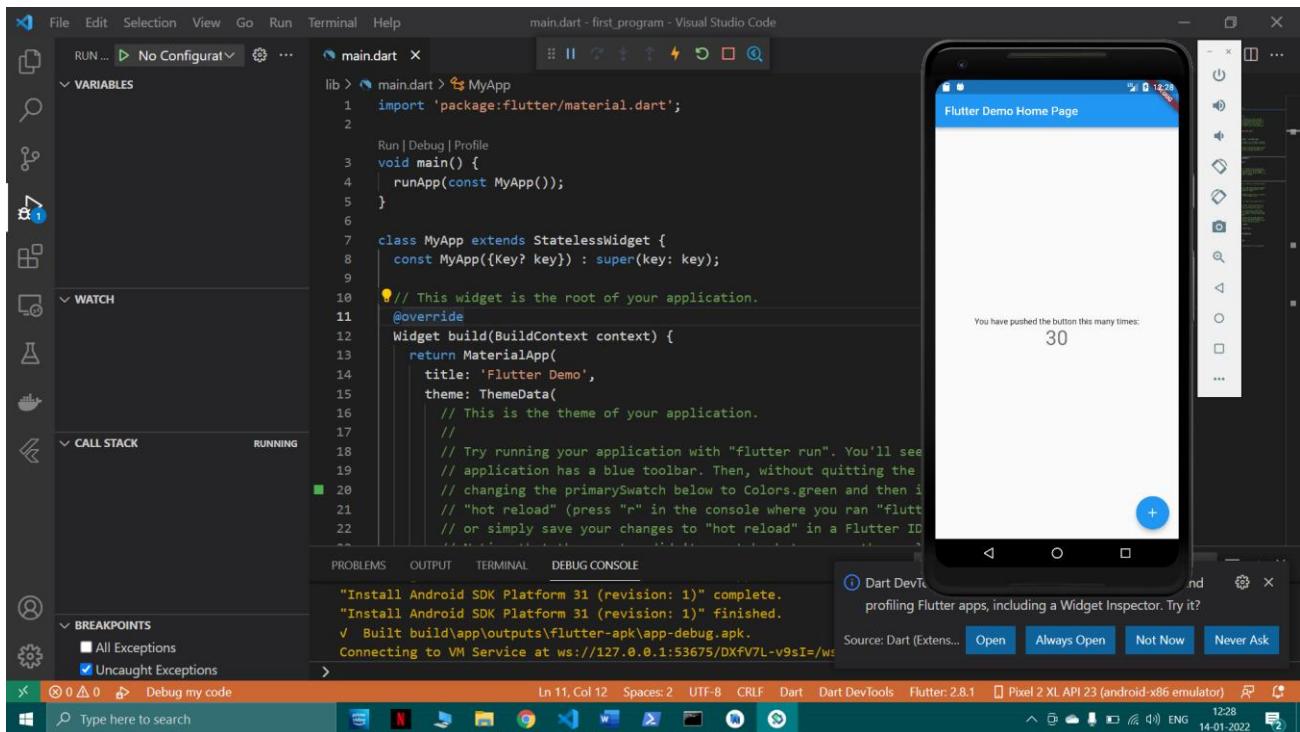
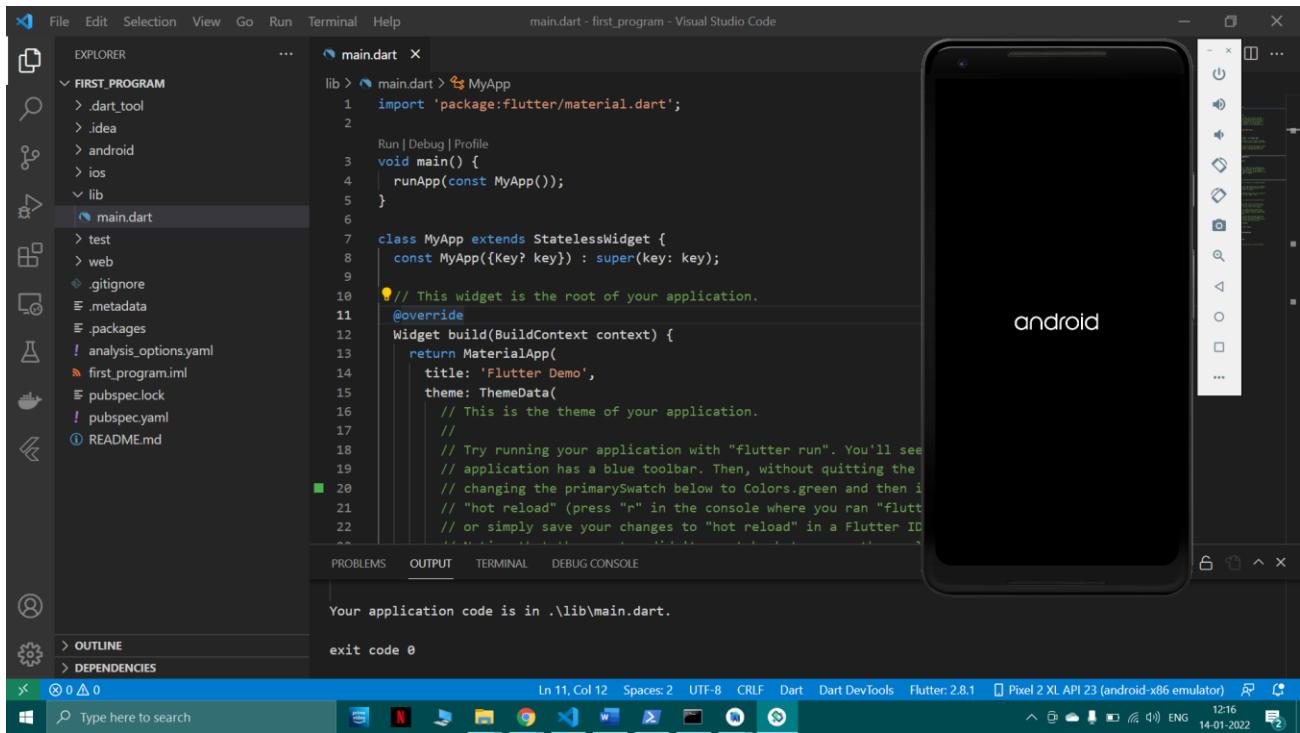
By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

```



8. Post-Experiments Exercise

A. Extended Theory:

1. Describe the Flutter project hierarchy in any IDE (Android Studio or VS Code).

Allan Rodrigues TE IT A-59 Rajdhani
DATE / /

Q.8

4)

1) Where

- 1) Idea - This folder holds configuration Android studio
- 2) Android - This folder contains a complete Android native app project & is used when building your flutter application for android

3) build - This folder is holding the compiled code of your flutter application. The content of this folder is automatically generated as part of the flutter build process.

4) iOS - This folder contains code for native project which is used when building your flutter app for iOS

5) lib - Inside lib folder you'll find Dart files which contain the code of your flutter application. By default it contains main.dart file.

6) test - The test folder is containing code which is written for flutter app in order to perform automated test when building.

7) .gitignore file : This is a text file containing a list of files, file extensions, folders which should be ignored while working with git

8) metadata - This file is managed by flutter automatically & is used to track properties of flutter project.

CS Scanned with CamScanner

Allan Rodrigues TE IT A-59

Rajdhani

DATE / /

- 9) Packages - This file contains automatically generated content by the flutter SDK
- 10) Pubspec.yaml - It contains name, description & version of the project, project dependencies and assets (images).

CS Scanned with CamScanner

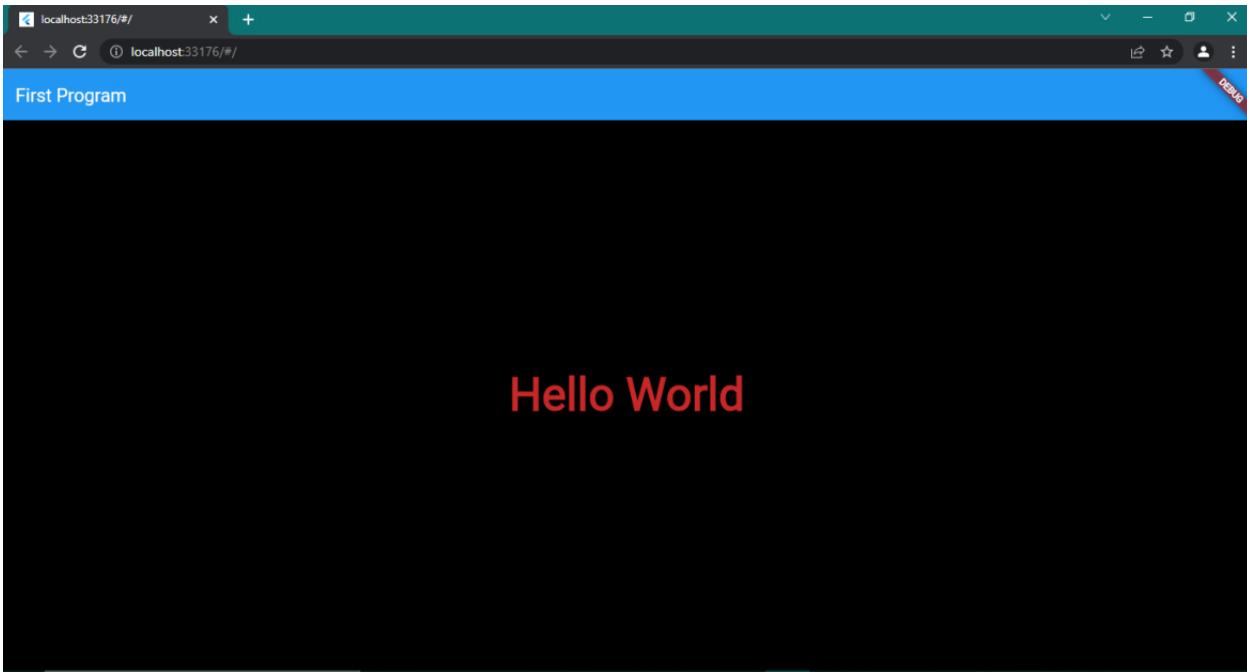
B. Questions:

- Execution of ‘Hello world’ program in Flutter.

The screenshot shows the Visual Studio Code interface with the 'helloworld.dart' file open in the editor. The code implements a 'GeeksForGeeks' StatelessWidget that returns a MaterialApp with a Scaffold containing a centered Text widget with bold red font and a black background.

```
lib > helloworld.dart > GeeksForGeeks > build
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(GeeksForGeeks());
5 }
6
7 class GeeksForGeeks extends StatelessWidget{
8   Widget build(BuildContext context){
9
10   // Material App
11   return MaterialApp(
12
13     // Scaffold Widget
14     home: Scaffold(
15       appBar: AppBar(
16
17         // AppBar takes a Text Widget
18         // in its title parameter
19         title: Text('GFG'),
20       ), // AppBar
21       body: Center(
22         child: Text('Hello World',
23           style: TextStyle(
24             fontSize: 50,
25             fontWeight: FontWeight.bold,
26             color: Colors.red[800],
27           ) // TextStyle
28         ), // Text
29         ), // Center
30         backgroundColor: Colors.black,
31       ) // Scaffold
32     ); // MaterialApp
33   }
34 }
```

This screenshot is identical to the one above, but it includes a red rectangular highlight around the closing brace '}' at the bottom of the code block, indicating the point where the user has just finished typing or is about to run the code.



C. Conclusion:

1. Write what was performed in the experiment.
2. Mention a few applications of what was studied.
3. Write the significance of the topic studied in the experiment.

c) Conclusion

In this experiment we installed flutter framework & Android studio. We set upon Android device using Android emulator. We created our first application using flutter.

Flutter allows you to create cross platform apps that provide native performance. Apps created with flutter feature beautiful & intuitive designs & are able to run animations smoothly.

CS Scanned with CamScanner

D. References:

1. <https://docs.flutter.dev/get-started/install>.
2. <https://www.javatpoint.com/flutter>
3. <https://www.javatpoint.com/flutter-dart-programming>