

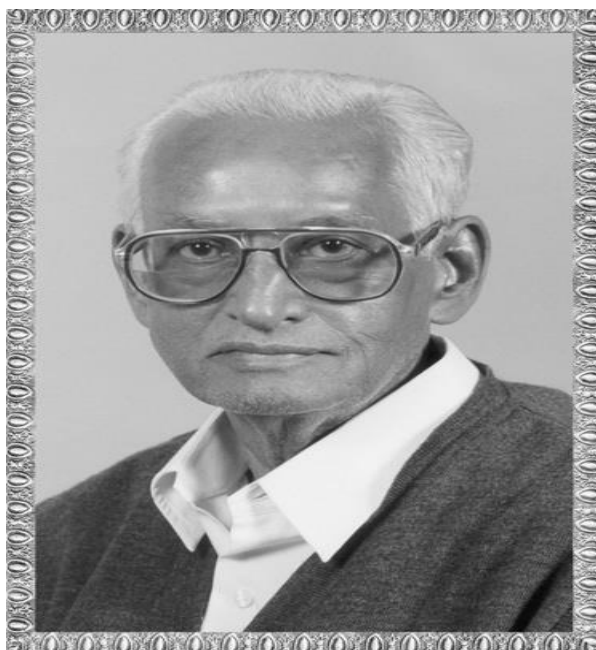
SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645



Lt. Shree Chimanbhai Shukla

MSCIT SEM-3 – FLUTTER APP DEVELOPMENT

Shree H.N.Shukla College
Street No. 2, Vaishali Nagar,
Nr. Amrapali Under Bridge,
Raiya Road, Rajkot.
Ph. (0281)2440478, 2472590



Shree H.N.Shukla College
Street No. 2, Vaishali Nagar,
Nr. Amrapali Under Bridge,
Raiya Road, Rajkot.
Ph. (0281)2440478, 2472590

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

CS – 13 : Flutter App Development		
SR.NO.	TOPIC	DETAIL
1	Introduction to Flutter and Dart	• Overview of Flutter and Dart • Overview of the Flutter architecture and how it works • Setting up the development environment • Dart syntax and data types • Basic Flutter widgets and layout • Basic Flutter widget properties and methods • Dart libraries and packages • Control flow and loops in Dart
2	Building User Interfaces	• Stateful vs. Stateless widgets • Layout and widgets hierarchy • Navigation and routing • Responsive design and media queries • Advanced Flutter widgets (e.g., SliverAppBar, AnimatedContainer) • Custom widget creation in Flutter • Debugging UI issues in Flutter • Advanced layout techniques (e.g., Flexbox, GridView)
3	Managing App State	• State management in Flutter • InheritedWidget and InheritedModel • ScopedModel and Provider • BLoC pattern for state management • Stream-based state management with RxDart

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

		• Redux architecture for Flutter • Firebase integration for state management • Using the Flutter DevTools for debugging
4	Networking and Persistence	• RESTful APIs and HTTP requests • JSON serialization and deserialization • SQLite and local storage • Shared preferences and secure storage • WebSockets for real-time communication in Flutter • Firebase integration for data storage • Caching data in Flutter • Using third-party libraries for networking and data storage (e.g., Dio, Hive)
5	Animation, Integration, Testing, Debugging & Accessibility	• Animations and motion • Advanced animation techniques (e.g., Flare, Lottie) • Internationalisation and localization • Native platform integration • Push notifications in Flutter (Firebase) • Integration with other native features (e.g., camera, location) • Testing and debugging • Accessibility in Flutter apps

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

CHAPTER-5

Animation, Integration, Testing, Debugging & Accessibility

- ❖ Animations and motion
- ❖ Internationalisation and localization
- ❖ Native platform integration
- ❖ Push notifications in Flutter (Firebase)
- ❖ Integration with other native features (e.g., camera, location)
- ❖ Testing and debugging
- ❖ Accessibility in Flutter apps

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Q-1 Explain Animations in Flutter.

Answer:

- ❖ Flutter provides http package to consume HTTP resources. http is a Future-based. The animation is a very powerful and important concept in Flutter.
- ❖ We cannot imagine any mobile app without animations.
- ❖ When you tap on a button or move from one page to another page are all animations.
- ❖ Animations enhance user experiences and make the applications more interactive.
- ❖ Flutter provides excellent support for animation and can separate the animation into two main categories, which are given below:
 - Tween Animation
 - Physics-based Animation

Tween Animation

- ❖ It is the short form of **in-betweening**.
- ❖ In a tween animation, it is required to define the **start** and **endpoint** of animation.
- ❖ It means the animation begins with start value, then goes through a series of intermediate values and finally reached the end value.
- ❖ It also provides the timeline and curve, which defines the time and speed of the transition.
- ❖ The widget framework provides a calculation of how to transition from the start and endpoint.

Example

```
ColorTween
{
  begin: color.green,
  end: color.blue,
}
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

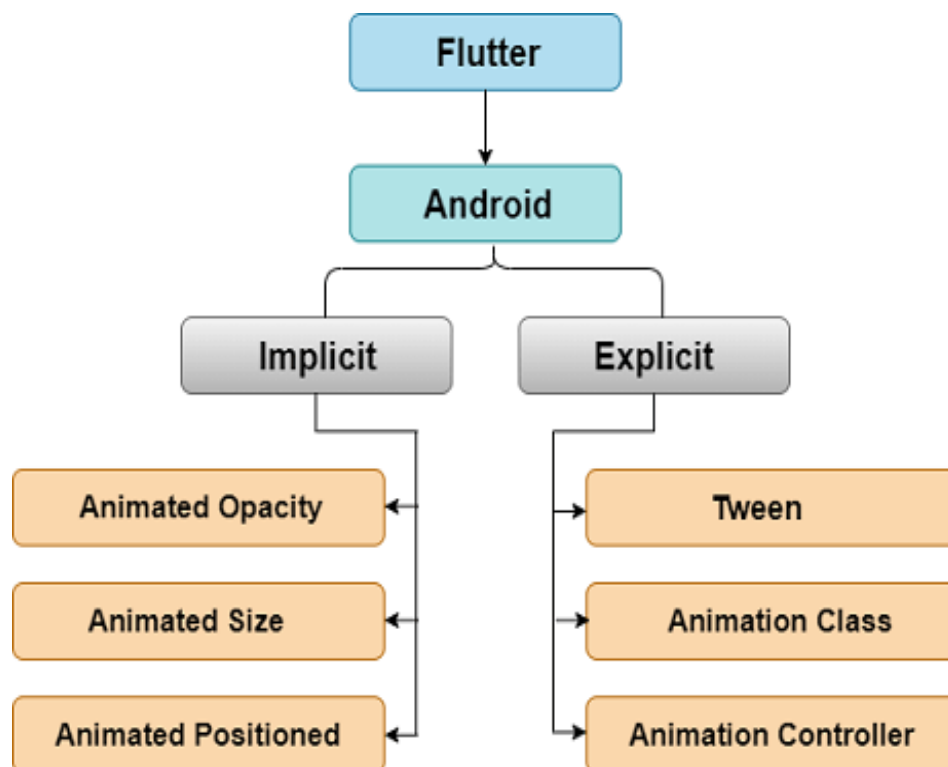
3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Physics-based Animation

- ❖ It is a type of animation which allows you to make an app interaction feels **realistic** and **interactive**.
- ❖ It simulates the real-world animation/movement, such as you want to animate a widget like spring, falling, or swinging with gravity.
- ❖ Thus, it is an animation that animates in response to user input/movement.
- ❖ The simplest example is the time of flight, and the distance of travel covered will be calculated according to the laws of physics.
- ❖ Flutter provides two types of techniques for animation. These techniques are:

1. Implicit Animation
2. Explicit Animation

- ❖ The following figure sets out the animation hierarchy in Flutter and explains it more clearly about the implicit and explicit animation.



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ Now, we are going to see how we can create explicit animation in Flutter.
- ❖ There are mainly three pillars of an animation, which are given below:

1. Ticker
2. Animation Class
3. AnimationController

Ticker

- ❖ The Ticker is a class which sends a signal at a regular interval, i.e., around 60 times per second.
- ❖ You can understand it with your watch, which ticks at regular intervals.
- ❖ At each tick, Ticker provides a callback method with the duration since the first ticks at each second, after it was started.
- ❖ Even if the tickers started at different times, it always synchronized automatically.
- ❖ The reason behind this is that the tickers give their elapsed time relative to the first tick after it was started.

Animation

- ❖ The Animation class is the core building block of the animation system.
- ❖ The animation is nothing else, but it represents a value (specific type) that can change over the lifetime of an animation.
- ❖ In Flutter, the widgets which perform an animation take an animation object as a parameter.
- ❖ This Animation object gives the information from which they read the current value of the animation and to which they listen for changes to that value.
- ❖ The animation class contains two methods **addListener()** and **addStatusListener()**.
- ❖ When the value of animation changes, it notifies all the listeners added with **addListener()**.
- ❖ Again, when the status of the animation changes, it notifies all the listeners added with **addStatusListener()**.
- ❖ The most common Animation classes are:

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- **Animation<double>**: It interpolates values between two decimal numbers over a certain duration.
- **Animation<Color>**: It interpolates colors between two color values.
- **Animation<Size>**: It interpolates sizes between two size values.

Animation Controller

- ❖ The animation controller is a class that allows us to control the animation.
- ❖ It always generates new values whenever the application is ready for a new frame.
- ❖ For example, it gives the controlling of start, stop, forward, or repeat of the animation.
- ❖ Once the animation controller is created, we can start building other animation based on it, such as reverse animation and curved animation.

animcontroller = AnimationController(vsync: this, duration: Duration(milliseconds: 2500));

- ❖ Here, the duration option controls the **duration** of the animation process, and vsync option is used to optimize the resource used in the animation.
- ❖ The basic steps necessary for using an AnimationController are:

Step 1: First, instantiate an AnimationController with parameters, such as duration and vsync.

Step 2: Add the required listeners like addListener() or addStatusListener().

Step 3: Start the animation.

Step 4: Perform the action in the listener callback methods (for example, setState).

Step 5: Last, dispose of the animation.

- ❖ Let us see a simple animation example, which uses an animation class and animation controller.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ The following example shows the **tween animation** that gives the start and endpoint of animation. Open the project and replace the following code in the **main.dart** file.

Q-2 Explain motion in Flutter.

Answer:

- ❖ Motion is basically a widget which handles initialization, state changes, code clean-ups and all the other animation boilerplate code, keeping you in the driver's seat by allowing you control the animations at each interval(literally).
- ❖ It's a real time-saver which takes care of animations and let's you concentrate on other important aspects of your app.

How to use it?

- ❖ **Row & Column** are two of the most basic widgets in Flutter used to design any UI. Motion Widget supports both these widgets and can be used as their replacement.

```
Motion<T>(  
  children:<Widget>[]  
)
```

- ❖ T type can be Row or Column. **children** can have a list of Flutter widgets as its value.
- ❖ Now the question is : How will they animate?
- ❖ This brings **MotionElement** into the picture.
- ❖ It is a custom widget which works hand-in-hand with the Motion Widget.
- ❖ It contains all the animation information for a child widget and tells the Motion Widget to animate it accordingly.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

```
Motion<T>(  
  children:<Widget>[  
    MotionElement(  
      child: Text("Hi")  
    ),  
    Text("How are you?")  
  ]  
)
```

- *The child widgets which are supposed to animate should be wrapped with MotionElement widget.*
- In the above code, Text widget which is the child of MotionElement will animate while the other one won't.

Motion & MotionElement :

- This concept revolves around the two widgets : **Motion** and **MotionElement**.

Motion

- ❖ This can be used as a Row or a Column and has all their properties. It also has parameters for controlling animation:
- **durationMs** : It accepts an integer which will be the animation duration (in milliseconds). By default, this value will be 1000.
- **shouldRepeat** : If we need to repeat the animation, we can set this property as true.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- **isAutomatic** : By default, motion widget will start animation automatically. We can control that using this flag.
- **exitConfigurations** : This accepts a **MotionExitConfigurations** object. Using this, we can configure exit animation. It has properties like **durationMs**, **displacement**, **orientation** and **mode**.
- This transition will be applied to the entire Motion Widget. *It provides you with a controller to control the exit animation.*

MotionElement

- ❖ This is used as a parent of any element inside Motion Widget which is supposed to animate. This has following properties to play with:
- **mode** : The is used to define mode of animation. Currently, Motion Widget supports **TRANSLATE**, **FADE** and **TRANSLATE_FADE**(translate + fade) modes.
- **orientation** : This defines the direction of the motion. It can be **LEFT**, **RIGHT**, **UP** or **DOWN**.
- **displacement** : It takes a double value which defines the translation value of the widget. This will be taken into consideration only if the widget is to be translated.
- **interval** : This property is very important when is comes to controlling animations granularly. It accepts Flutter's **Interval** object. We can mention begin and end time

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

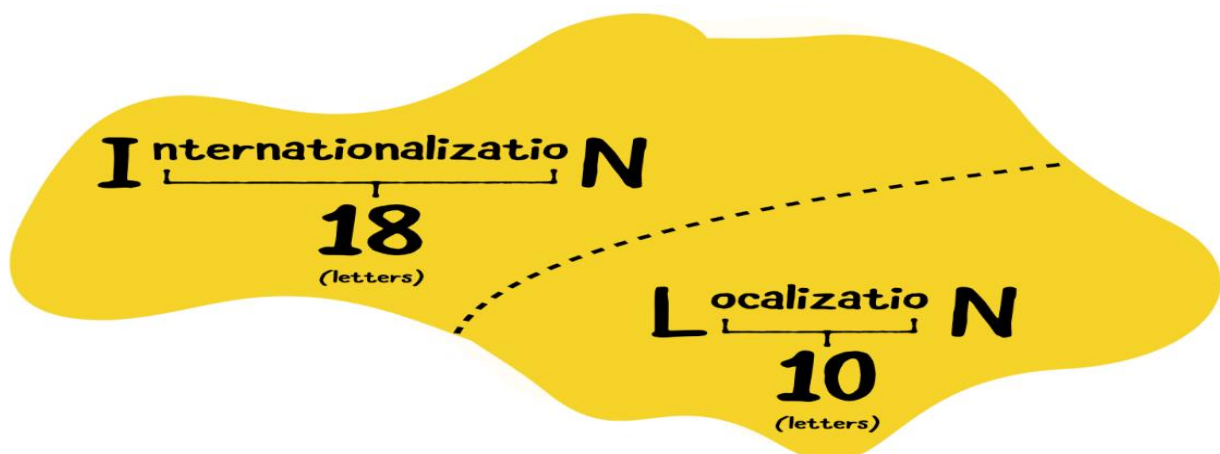
3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

values to delay the animation as per the needs. The values should be between 0.0 and 1.0. We can also specify curves to have complex interpolations.

Q-3 Explain Internationalisation and localization in Flutter.

Answer:

- ❖ Localization is the process of making your app available to other locales.
- ❖ It includes the parameters defining the user's language, region and any particular variant — for example, en-US (American English).
- ❖ Internationalization is how you architect and engineer your code to make it easy to localize.
- ❖ So, yes, you can have an internationalized app that supports just one locale, because it has the capacity to ultimately support more than one locale without dramatic engineering efforts. In fact, that's encouraged.
- ❖ Nowadays, mobile applications are used by customers from different countries and as a result, applications are required to display the content in different languages.
- ❖ Enabling an application to work in multiple languages is called Internationalizing the application.
- ❖ For an application to work in different languages, it should first find the current locale of the system in which the application is running and then need to show it's content in that particular locale, and this process is called Localization.



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ Internationalization, a.k.a. i18n, is the design and development of a Flutter application that provides its easy localization for users around the globe.
- ❖ Localization is the process of adapting an internationalized app for a specific region or locale, including translating text and adding locale-specific functionality.
- ❖ Translation is the process of changing a text into another language, and it is just one aspect of the localization process.

Flutter localizations package

- ❖ The `flutter_localizations` package comes with Flutter and it contains several localization options for Flutter widgets.
- ❖ See the [Flutter documentation](#) to see what localization settings are available with the package.
- ❖ **Intl library** is the most important library on the market for Flutter i18n that is an official Dart package.
- ❖ The `Intl` class is set with the default locale and many methods to access most of the i18n mechanisms.
- ❖ Some of the i18n and l10n facilities available in intl are `DateFormat`, `NumberFormat`, `BidiFormatter` classes, plurals, genders, and most importantly, message translations.
- *Localizations* – Localizations is a generic widget used to set the Locale and the localized resources of its child.

```
class CustomLocalizations {  
  CustomLocalizations(this.locale);  
  final Locale locale;  
  static CustomLocalizations of(BuildContext context) {  
    return Localizations.of<CustomLocalizations>(context, CustomLocalizations);  
  }  
}
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

```
static Map<String, Map<String, String>> _resources = {  
  'en': {  
    'title': 'Demo',  
    'message': 'Hello World'  
  },  
  'es': {  
    'title': 'Manifestación',  
    'message': 'Hola Mundo',  
  },  
};  
String get title {  
  return _resources[locale.languageCode]['title'];  
}  
String get message {  
  return _resources[locale.languageCode]['message'];  
}  
}
```

Using intl Package

- ❖ Flutter provides intl package to further simplify the development of localized mobile application.
 - ❖ intl package provides special methods and tools to semi-auto generate language specific messages.
 - ❖ Let us create a new localized application by using intl package and understand the concept.
- Create a new flutter application, flutter_intl_app.
 - Open pubspec.yaml and add the package details.

dependencies:

flutter:

sdk: flutter

flutter_localizations:

sdk: flutter

intl: ^0.15.7

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

intl_translation: ^0.17.3

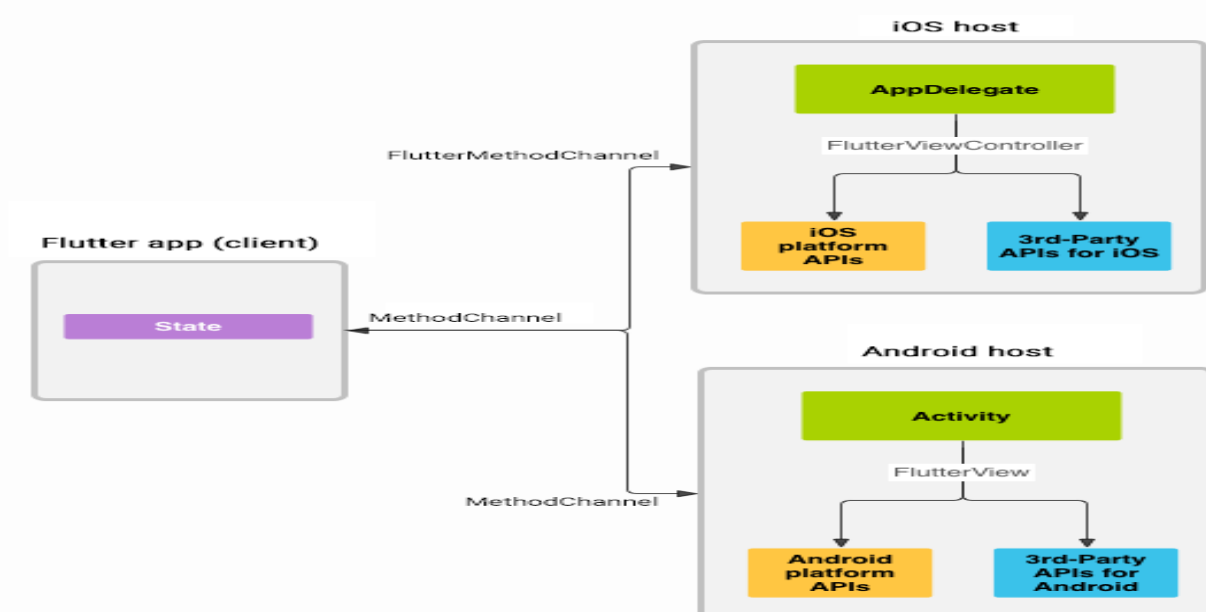
Q-4 Explain Native Platform Integration in Flutter.

Answer:

- ❖ Flutter is a cross-platform toolkit, yet it allows you the full power of native platform APIs in a very simple way.
- ❖ Even if you don't have any past experience writing native code.
- ❖ we will explore platform channels in Flutter and how they can be used to interact with platform-specific features or native APIs.

What are Platform Channels?

- ❖ Platform channels are a feature in Flutter that allows developers to interact with platform-specific code.
- ❖ With platform channels, developers can send messages from their Flutter app to the native code on the platform, and vice versa. This makes it possible to access platform-specific features that are not available through Flutter's core APIs.



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



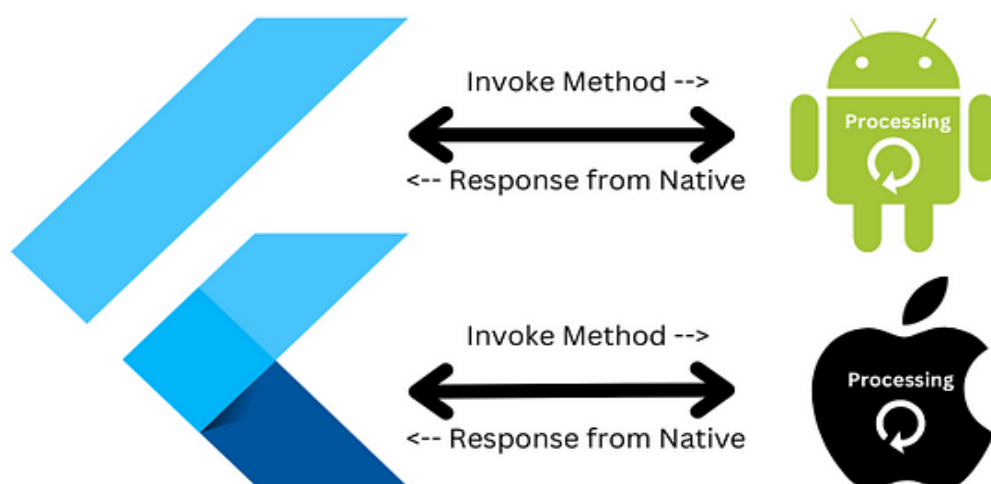
2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Why Platform Channels?

- ❖ Let's imagine a common use-case. A client wants to integrate with a 3rd-party provider.
- ❖ This provider doesn't have support for Flutter yet, but they have an iOS and Android SDK.
- ❖ In such case, you will need to call these SDKs from Flutter.
- ❖ For example, you may want to access the device's camera or microphone, or access a native library that is not available in Flutter.
- ❖ Platform channels make it possible to access these features by creating a bridge between the Flutter app and the native code on the platform.

Platform Channel Bridge



SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

There are 2 types of platform channels:

- [MethodChannel](#): a channel used to communicate with the platform using asynchronous method calls. Think of it as a Future.
- [EventChannel](#): a channel used to listen to event streams from the platform. Think of it as a Stream.

MethodChannel

Step 1: Create the Flutter platform client

1. Once you have a MethodChannel object ready, you can call any method by its name:

Step 2: Android platform-specific implementation

1. Start android Studio
2. Open the file MainActivity.kt located in the my_flutter_app/android/app/src/main/kotlin/ **folder** in the Project view.

Step 3: iOS platform-specific implementation

1. Start Xcode
 2. Open ios folder in your flutter project in Xcode.
 3. Open the file AppDelegate.swift located under **Runner > Runner** in the Project navigator.
 4. Override the application:didFinishLaunchingWithOptions: function and create a FlutterMethodChannel tied to the channel name app.id.com/my_channel_name.
- ❖ When the native code receives the message, it can perform the necessary actions and send a response back to the Flutter app.
 - ❖ In this case, the native code would return the battery level as an integer, which is then converted to a string and displayed in the UI.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Q-5 Explain Push notifications in Flutter (Firebase).

Answer:

- ❖ In today's world, mobile applications have become an integral part of our daily lives.
- ❖ These applications have made our lives easier and more comfortable.
- ❖ Push notifications have emerged as a crucial feature in mobile applications that keep users updated and engaged.
- ❖ Flutter, Google's mobile app development framework, is an excellent platform for developing cross-platform mobile applications.
- ❖ Firebase Cloud Messaging is a free service provided by Google that allows developers to send push notifications to Android, iOS, and web applications.
- ❖ It is a powerful tool that allows developers to engage their users and improve their application's retention rates.
- ❖ FCM provides a simple and easy-to-use API that developers can use to send notifications to their users.
- ❖ Firebase also provides an easy integration with Flutter applications.
- ❖ Now let's start setting up the push notifications using Firebase Cloud Messaging in Flutter applications.

Step 1:

- ❖ Create a Firebase project To use Firebase services, you need to create a Firebase project. Go to the Firebase Console and create a new project.
- ❖ Follow the instructions provided by Firebase to create a new project.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Step 2:

- ❖ Add Firebase to your Flutter project After creating a Firebase project, you need to add the Firebase SDK to your Flutter project.
- ❖ To add Firebase to your Flutter project, you need to follow these steps:
 1. Go to the Firebase Console and click on the “Add Firebase to your Android/iOS app” button.
 2. Provide the required information, such as your package name and app nickname, to generate the configuration files.
 3. Download the configuration files and add them to your Flutter project.

Step 3:

- ❖ Add the Firebase Cloud Messaging dependency After adding Firebase to your Flutter project, you need to add the Firebase Cloud Messaging dependency to your project.
- ❖ To add the Firebase Cloud Messaging dependency, add the following line to your project's pubspec.yaml file:

Step 4:

- ❖ Configure Firebase Cloud Messaging After adding the Firebase Cloud Messaging dependency, you need to configure Firebase Cloud Messaging in your Flutter project.
- ❖ To configure Firebase Cloud Messaging, you need to follow these steps:

1. Add the following code to your main.dart file to initialize Firebase:

This code initializes Firebase Cloud Messaging and provides two functions to get the token and listen to incoming notifications.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Step 5:

- ❖ Test the push notifications After configuring Firebase Cloud Messaging, you can test the push notifications in your Flutter application.
- ❖ To test the push notifications, follow these steps:
 1. Start your Flutter application and get the token by calling the `getToken()` function.
 2. Go to the Firebase Console and select your project.
 3. Click on the “Cloud Messaging” tab and click on the “New Notification” button.
 4. Provide the required information, such as the title and body of the notification, and click on the “Send Test Message” button.

Step 6:

- ❖ Handle notifications in the background or killed mode When the application is in the background or killed mode, the push notifications are handled differently than when the application is in the foreground.
 - ❖ In this step, we will handle the notifications when the application is in the background or killed mode.
1. Add the following code to the `PushNotificationService` class to listen to incoming notifications when the application is in the background or killed mode:

```
Future<void> backgroundHandler(RemoteMessage message) async {  
  print('Handling a background message ${message.messageId}');  
}
```

2. Modify the `initialize` function in the `PushNotificationService` class to include the `backgroundHandler` function:

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

```
Future initialize() async {  
  FirebaseMessaging.onMessage.listen((RemoteMessage message) {  
    print('Got a message whilst in the foreground!');  
    print('Message data: ${message.data}');  
  
    if (message.notification != null) {  
      print('Message also contained a notification: ${message.notification}');  
    }  
  });  
  
  FirebaseMessaging.onBackgroundMessage(backgroundHandler);  
  
  // Get the token  
  await getToken();  
}
```

3.Add the following code to the AndroidManifest.xml file to enable the background handling of notifications:

```
<service  
  android:name="com.example.myapp.PushNotificationService"  
  android:exported="false">  
  <intent-filter>  
    <action android:name="com.google.firebase.MESSAGING_EVENT" />  
  </intent-filter>  
</service>
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

4. To handle the notifications when the application is in the background or killed mode, add the following code to the `onLaunch` and `onResume` functions of the `MaterialApp` widget:

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp();  
  FirebaseMessaging.onBackgroundMessage(backgroundHandler);  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  final PushNotificationService _notificationService = PushNotificationService();  
  
  @override  
  Widget build(BuildContext context) {  
    _notificationService.initialize();  
  
    return MaterialApp(  
      title: 'My App',  
      home: MyHomePage(),  
      onGenerateRoute: Router.generateRoute,  
      initialRoute: '/',  
      onLaunch: (Map<String, dynamic> message) async {  
        print('onLaunch: $message');  
        // Handle the notification  
      },  
      onResume: (Map<String, dynamic> message) async {  
        print('onResume: $message');  
        // Handle the notification  
      },  
    );  
  }  
}
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Q-6 Explain Integration with other native features (e.g., camera, location) in Flutter.

Answer:

1. Camera :-

- ❖ Many apps require working with the device's cameras to take photos and videos.
- ❖ Flutter provides the camera plugin for this purpose.
- ❖ The camera plugin provides tools to get a list of the available cameras, display a preview coming from a specific camera, and take photos or videos.

- ❖ This recipe demonstrates how to use the camera plugin to display a preview, take a photo, and display it using the following steps:
 1. Add the required dependencies.
 2. Get a list of the available cameras.
 3. Create and initialize the CameraController.
 4. Use a CameraPreview to display the camera's feed.
 5. Take a picture with the CameraController.
 6. Display the picture with an Image widget.

1. Add the required dependencies

- ❖ To complete this recipe, you need to add three dependencies to your app: camera
- ❖ Provides tools to work with the cameras on the device.

path_provider
- ❖ Finds the correct paths to store images.

path

Shree H.N.Shukla College of I.T & Management

"Sky is the Limit"

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ Creates paths that work on any platform.

2. Get a list of the available cameras

- ❖ Next, get a list of available cameras using the camera plugin.

content_copy

```
// Ensure that plugin services are initialized so that `availableCameras()`  
// can be called before `runApp()`  
WidgetsFlutterBinding.ensureInitialized();  
  
// Obtain a list of the available cameras on the device.  
final cameras = await availableCameras();  
  
// Get a specific camera from the list of available cameras.  
final firstCamera = cameras.first;
```

3. Create and initialize the CameraController

- ❖ Once you have a camera, use the following steps to create and initialize a CameraController.
- ❖ This process establishes a connection to the device's camera that allows you to control the camera and display a preview of the camera's feed.
 1. Create a StatefulWidget with a companion State class.
 2. Add a variable to the State class to store the CameraController.
 1. Add a variable to the State class to store the Future returned from CameraController.initialize().
 3. Create and initialize the controller in the initState() method.
 4. Dispose of the controller in the dispose() method.

5. Use a CameraPreview to display the camera's feed

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ Next, use the CameraPreview widget from the camera package to display a preview of the camera's feed.

6. Take a picture with the CameraController

- ❖ You can use the CameraController to take pictures using the takePicture() method, which returns an XFile, a cross-platform, simplified File abstraction.
- ❖ On both Android and IOS, the new image is stored in their respective cache directories, and the path to that location is returned in the XFile.
- ❖ In this example, create a FloatingActionButton that takes a picture using the CameraController when a user taps on the button.

- ❖ Taking a picture requires 2 steps:

1. Ensure that the camera is initialized.
2. Use the controller to take a picture and ensure that it returns a Future<XFile>.

7. Display the picture with an Image widget

- ❖ If you take the picture successfully, you can then display the saved picture using an Image widget.
- ❖ In this case, the picture is stored as a file on the device.
- ❖ Therefore, you must provide a File to the Image.file constructor.
- ❖ You can create an instance of the File class by passing the path created in the previous step.

content_copy

```
Image.file(File('path/to/my/picture.png'));
```

2. Location :-

- ❖ Nowadays Google Maps is one of the popular Applications for navigation or finding any Location.
- ❖ With this, we can find out the current Location of any person. So , we are going to see how to get users' current Location in Flutter.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Step 1: Create a New Project in Android Studio

- ❖ To set up Flutter Development on Android Studio please refer to **Android Studio Setup for Flutter Development**, and then create a new project in Android Studio please refer to **Creating a Simple Application in Flutter**.

Step 2: Generating API Key for Using Google Maps

- ❖ Check out the below article on **How to generate an API key for using Google Maps**. After developing that key we have to use it within our project to integrate Google Maps.
- ❖ *Now, follow the Steps to Implement Google Maps in the Flutter Application*

Step 3: Adding Dependency to pubspec.yaml File

dependencies:

flutter:

google_maps_flutter: ^2.1.8

geolocator: ^9.0.2

Now click on **pub.get** to configure.

Step 4: Update build.gradle File

- ❖ Navigate to android>app>build.gradle file and update code on the below line.

Change Compile SDK Version

android {

compileSdkVersion 31

compileOptions {

sourceCompatibility JavaVersion.VERSION_1_8

targetCompatibility JavaVersion.VERSION_1_8 }

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Step 5: Add API Key to Android Manifest File

- ❖ Navigate to **android > app > src > main > AndrodManifest.xml** file and add the below code in the manifest tag
- ❖ . Make sure to add your Google Maps API key in the value.

```
<meta-data android:name="com.google.android.geo.API_KEY"
            android:value="Enter your API key here"/>
```

Step 6: Add the Following Permissions to the Android Manifest File

```
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission
    android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
```

Step 7: Working with main.dart File

- ❖ Navigate to **lib > main.dart** file and add the below code to it
- ❖ . Comments are added in the code to get to know in detail.

- Dart
- `import 'package:flutter/material.dart';`
- `import 'package:google_maps/HomePage.dart';`

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

```
void main() {  
  runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  // This widget is the root of your application.  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      // on below line we are specifying title of our app  
      title: 'GFG',  
      // on below line we are hiding debug banner  
      debugShowCheckedModeBanner: false,  
      theme: ThemeData(  
        // on below line we are specifying theme  
        primarySwatch: Colors.green,  
      ),  
      // First screen of our app  
      home: const HomePage(),  
    );  
  }  
}
```

Step 8: Creating a new Dart File for Home Page

- ❖ Navigate to **lib** > **Right click on it** > **New** > **Dart File** and name it as **HomePage.dart**. After creating that file add the below code to it.

**3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645**



(1) Testing :-

- ❖ Testing and Debugging Flutter Apps is critical to their app development process.
- ❖ Their primary struggle when entering the depths of application creation is testing and debugging Flutter apps.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Different Types Of Tests In The World Of Flutter Apps

- ❖ A variety of tests can be performed on a flutter application depending on what aspect you want to validate:
- ❖ **Unit Tests:** These are used for verifying single functions or methods in isolation.
- ❖ **Widget Tests:** An integral part when developing UIs using flutter widgets which helps ensure they work together seamlessly.
- ❖ **Integration Test:** This type checks whether all parts interact smoothly without causing any disruption or errors during operation.

Manual Testing with Real Device Cloud

- ❖ A common approach for developers when doing app development is manual testing on a real device cloud.
- ❖ This involves deploying your application onto physical devices under different environments - replicating how users interact with the product.
- ❖ It provides invaluable insights into performance and user experience.

Automated Tests Using Flutter Driver

- ❖ An alternative strategy uses an automation framework called '[Flutter Driver](#).'
- ❖ With this tool, actions such as button presses or form submissions can be scripted automatically to verify your app's correct responses without human intervention.
- ❖ This **frees up valuable developer resources** while ensuring consistent test coverage across all functionalities.

Unit Tests: The Building Blocks

- ❖ They focus on single methods or classes, ensuring they return expected results under varying conditions.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Friendly Interactions via Widget Tests

- ❖ Widget testing, exclusive to Flutter applications, examines interactions between widgets (the building blocks for any UI).
- ❖ These simulate user interaction patterns such as tapping buttons or scrolling down pages and validate if these actions trigger desired responses from the system.

Integration Tests

- ❖ An integration test, or an end-to-end system test, evaluates how multiple units work together by running complete scenarios similar to what users would do while using the app in real-life situations.
- ❖ Remember that each type has its place within a *comprehensive testing strategy*; choosing which one(s) best suits your needs depends on factors like project scope and complexity.
- ❖ The value proposition? Automated testing not only helps detect bugs early but also aids developers during refactoring stages, where changes might affect existing functionalities unintentionally.

(2) Debugging :-

- ❖ The craft of debugging is an essential component of the app creation procedure.
- ❖ Several tools and techniques can assist you in effectively identifying and resolving issues in Flutter apps.

Setting Breakpoints

- ❖ A common technique used in debugging any software or app is setting breakpoints.
- ❖ A breakpoint, as its name suggests, allows your code execution to pause at specific points where you suspect an issue might be occurring.
- ❖ This enables developers to inspect variables' values over time or understand why certain functions aren't behaving as expected.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Incorporating Log Messages

- ❖ Besides using breakpoints, logging messages are another effective method for diagnosing problems within an app's architecture.
- ❖ Integrating log statements throughout your Dart programming, language-based codebase makes it easier to track down when things start going awry during runtime.
- ❖ Flutter's developer library offers comprehensive logging capabilities that every developer should use while debugging their applications built by **TechAhead**.

Error Handling with Try-Catch Blocks

- ❖ Last but not least on our list are error-handling mechanisms such as *try-catch blocks*; these play a vital role in identifying errors before they crash your entire application completely out of the blue.
- ❖ Implementing appropriate error handlers helps maintain stability despite unexpected situations arising during runtime.
- ❖ While these methods prove highly useful for detecting bugs early in the development cycle, they do not replace thorough testing strategies, ensuring overall quality and reliability when building Flutter applications.

DevTools

- ❖ For debugging and profiling apps, DevTools might be the first tool you reach for.
- ❖ DevTools runs in a browser and supports a variety of features:
 - source-level debugger
 - widget inspector that displays a visual widget tree, and “widget select” mode where you select a widget in the app and it drills down to that widget in the tree
 - memory profiler
 - timeline view that supports tracing, and importing and exporting trace information
 - logging view

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- ❖ If you run your application in debug mode or profile mode, while it's running you can open DevTools in the browser to connect to your app.
- ❖ DevTools doesn't work well with an app compiled to release mode, as the debugging and profiling information has been stripped away.

Debugging application layers

- ❖ Flutter was designed with a layered architecture that includes widget, rendering, and painting layers.
- ❖ For links to more information and videos, see The Framework architecture on the GitHub wiki, and the community article, The Layer Cake.
- ❖ The Flutter widget inspector provides a visual representation of the widget tree, but if you want a greater level of detail, or you want a verbose text-based dump of the widget, layer, or render trees, see Debug flags: application layers in the Debugging Flutter apps programmatically page.

Debugging animations

- ❖ The easiest way to debug animations is to slow them down.
- ❖ The Flutter inspector provides a **Slow Animations** button, or you can slow the animations programmatically.
- ❖ For more information on debugging janky (non-smooth) applications, see Flutter performance profiling.

Q-8 Explain Accessibility in Flutter apps.

Answer :

- ❖ Flutter is committed to supporting developers in making their apps more accessible, and includes first-class framework support for accessibility in addition to that provided by the underlying operating system, including:

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Large fonts

Render text widgets with user-specified font sizes

Screen readers

Communicate spoken feedback about UI contents

Sufficient contrast

Render widgets with colors that have sufficient contrast

Inspecting accessibility support

- ❖ In addition to testing for these specific topics, we recommend using automated accessibility scanners:
- **For Android:**
 1. Install the Accessibility Scanner for Android
 2. Enable the Accessibility Scanner from **Android Settings > Accessibility > Accessibility Scanner > On**
 3. Navigate to the Accessibility Scanner ‘checkbox’ icon button to initiate a scan.
- **For iOS:**
 1. Open the iOS folder of your Flutter app in Xcode
 2. Select a Simulator as the target, and click **Run** button
 3. In Xcode, select **Xcode > Open Developer Tools > Accessibility Inspector**
 4. In the Accessibility Inspector, select **Inspection > Enable Point to Inspect**, and then select the various user interface elements in running Flutter app to inspect their accessibility attributes
 5. In the Accessibility Inspector, select **Audit** in the toolbar, and then select **Run Audit** to get a report of potential issues.
- **For web:**
 1. Open Chrome DevTools (or similar tools in other browsers)
 2. Inspect the HTML tree containing the ARIA attributes generated by Flutter.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

3. In Chrome, the “Elements” tab has a “Accessibility” sub-tab that can be used to inspect the data exported to semantics tree

Large fonts

- ❖ Both Android and iOS contain system settings to configure the desired font sizes used by apps.
- ❖ Flutter text widgets respect this OS setting when determining font sizes.
- ❖ Font sizes are calculated automatically by Flutter based on the OS setting.
- ❖ However, as a developer you should make sure your layout has enough room to render all its contents when the font sizes are increased.
- ❖ For example, you can test all parts of your app on a small-screen device configured to use the largest font setting.

Screen readers

- ❖ For mobile, screen readers (TalkBack, VoiceOver) enable visually impaired users to get spoken feedback about the contents of the screen and interact with the UI by using gestures on mobile and keyboard shortcuts on desktop.
- ❖ Turn on VoiceOver or TalkBack on your mobile device and navigate around your app.

To turn on the screen reader on your device, complete the following steps:

- TalkBack on Android
- VoiceOver on iPhone
- Browsers
- Desktop

- 1. On your device, open Settings.**
- 2. Select Accessibility and then TalkBack.**
- 3. Turn ‘Use TalkBack’ on or off.**
- 4. Select Ok.**

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

Sufficient contrast

- ❖ Sufficient color contrast makes text and images easier to read.
- ❖ Along with benefitting users with various visual impairments, sufficient color contrast helps all users when viewing an interface on devices in extreme lighting conditions, such as when exposed to direct sunlight or on a display with low brightness.
- At least 4.5:1 for small text (below 18 point regular or 14 point bold)
- At least 3.0:1 for large text (18 point and above regular or 14 point and above bold)

Testing accessibility on mobile

- ❖ Test your app using Flutter's Accessibility Guideline API.
- ❖ This API checks if your app's UI meets Flutter's accessibility recommendations.
- ❖ These cover recommendations for text contrast, target size, and target labels.
- ❖ The following example shows how to use the Guideline API on Name Generator.
- ❖ You created this app as part of the Write your first Flutter app codelab.
- ❖ Each button on the app's main screen serves as a tappable target with text represented in 18 point.

```
final SemanticsHandle handle = tester.ensureSemantics();  
await tester.pumpWidget(MyApp());
```

```
// Checks that tappable nodes have a minimum size of 48 by 48 pixels  
// for Android.  
await expectLater(tester, meetsGuideline(androidTapTargetGuideline));
```

```
// Checks that tappable nodes have a minimum size of 44 by 44 pixels  
// for iOS.  
await expectLater(tester, meetsGuideline(iOSTapTargetGuideline));
```

```
// Checks that touch targets with a tap or long press action are labeled.  
await expectLater(tester, meetsGuideline(labeledTapTargetGuideline));
```

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

```
// Checks whether semantic nodes meet the minimum text contrast levels.  
// The recommended text contrast is 3:1 for larger text  
// (18 point and above regular).  
await expectLater(tester, meetsGuideline(textContrastGuideline));  
handle.dispose();
```

Accessibility release checklist

- ❖ Here is a non-exhaustive list of things to consider as you prepare your app for release.
- **Active interactions :**
 - ✓ Ensure that all active interactions do something.
 - ✓ Any button that can be pushed should do something when pushed.
 - ✓ For example, if you have a no-op callback for an onPressed event, change it to show a SnackBar on the screen explaining which control you just pushed.
- **Screen reader testing:**
 - ✓ The screen reader should be able to describe all controls on the page when you tap on them, and the descriptions should be intelligible.
 - ✓ Test your app with TalkBack (Android) and VoiceOver (iOS).
- **Contrast ratios:**
 - ✓ We encourage you to have a contrast ratio of at least 4.5:1 between controls or text and the background, with the exception of disabled components.
 - ✓ Images should also be vetted for sufficient contrast.
- **Context switching:**
 - ✓ Nothing should change the user's context automatically while typing in information.
 - ✓ Generally, the widgets should avoid changing the user's context without some sort of confirmation action.

SHREE H. N. SHUKLA COLLEGE OF I.T. & MGMT.

(AFFILIATED TO SAURASHTRA UNIVERSITY)



2 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot – 360001
Ph.No–(0281)2440478

3 – Vaishalinagar
Nr. Amrapali Under Bridge
Raiya Road
Rajkot - 360001
Ph.No–(0281)2471645

- **Tappable targets:**

- ✓ All tappable targets should be at least 48x48 pixels.

- **Errors:**

- ✓ Important actions should be able to be undone. In fields that show errors, suggest a correction if possible.

- **Color vision deficiency testing:**

- ✓ Controls should be usable and legible in colorblind and grayscale modes.

- **Scale factors:**

- ✓ The UI should remain legible and usable at very large scale factors for text size and display scaling.