<u>Flutter</u>

Basic Questions (1-25)

1. **What is Flutter?**
   - Flutter is an open-source UI framework developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.
2. **What language is used to develop Flutter applications?**
   - Flutter applications are written in Dart, a language developed by Google, which is easy to learn and designed for building fast, high-performance apps.
3. **What is a widget in Flutter?**
   - In Flutter, everything is a widget. A widget is a description of the part of the user interface (UI) you want to build. Widgets are either stateless or stateful.

DM TECHPARK

4. **What is the difference between a StatelessWidget and a StatefulWidget?**
   - A `StatelessWidget` is immutable, meaning its state cannot change once it is created. A `StatefulWidget` has a mutable state, meaning its state can change during the lifetime of the widget.
5. **What is the role of the `build()` method in Flutter?**
   - The `build()` method in Flutter is called whenever the framework needs to render the widget or when the widget’s state changes. It describes how to display the widget.
6. **What is a Flutter App's entry point?**
   - The entry point of a Flutter application is the `main()` function, where the app is initialized and the root widget is set, usually

using `runApp()`.

7. **What is the purpose of the `MaterialApp` widget?**
   - The `MaterialApp` widget is a wrapper for the app, setting up the basic structure and configuration, such as routing, themes, and localization.

8. **What is hot reload in Flutter?**
   - Hot reload allows developers to see the changes they make in the code immediately without restarting the app, speeding up development.

9. **What is the difference between `Container` and `SizedBox` in Flutter?**
   - `Container` is a versatile widget used for creating layout elements, applying styles like padding, margin, decoration, etc. `SizedBox` is used to create a box with specific dimensions and can also be used for spacing between widgets.

IDM TECHPARK
GUIDE'S FOR PERFECT CAREER PATHWAY

10. **What is a Flutter `Scaffold`?**
    - `Scaffold` is a layout structure in Flutter that provides standard UI elements like `AppBar`, `Drawer`, `FloatingActionButton`, and a body area to place widgets.

11. **How does Flutter handle rendering of widgets?**
    - Flutter uses its own rendering engine (Skia) to draw widgets on the screen. It does not rely on native components, and instead, it uses its own framework to render everything.

12. **What is the `Navigator` widget used for in Flutter?**
    - The `Navigator` widget is used for navigating between different screens or pages in a Flutter app. It handles push and pop

operations in the stack of routes.

13. **What is a `Future` in Dart?**

  ○ A `Future` represents a value that is not available yet but will be at some point in the future, commonly used for asynchronous operations.

14. **What is a `Stream` in Dart?**

  ○ A `Stream` represents a sequence of asynchronous events, used when multiple values need to be handled over time.

15. **What is a `StatefulWidget` and when do you use it?**

  ○ A `StatefulWidget` is a widget that has mutable state, which means its UI can change during the lifetime of the widget. You use it when your widget needs to update dynamically.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

16. **What are keys in Flutter?**

  ○ Keys are used in Flutter to preserve the state of widgets when they move around in the widget tree, especially in cases of reordering lists or navigating.

17. **What is `setState()` in Flutter?**

  ○ `setState()` is a method used in StatefulWidgets to notify the framework that the widget's state has changed, causing the widget to rebuild.

18. **What is the difference between `build()` and `rebuild()` in Flutter?**

  ○ `build()` is called when the widget is created or when the widget tree is first rendered. `rebuild()` is not a term used in Flutter, but the `build()` method is called whenever the widget's state changes or when the widget’s parent widget rebuilds.

19. **What is the purpose of the `Text` widget in Flutter?**

    - The `Text` widget is used to display a string of text in a Flutter application. It supports customization like font size, color, and style.

20. **What is the `pubspec.yaml` file in Flutter?**

    - `pubspec.yaml` is the configuration file for a Flutter project, where you define dependencies, assets, fonts, and other configuration settings.

21. **What is a `ListView` in Flutter?**

    - `ListView` is a scrollable list of widgets that allows you to display multiple items in a list. It is commonly used when the list length is dynamic.

DM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

22. **How does Flutter manage state?**

    - Flutter provides multiple ways to manage state, including using `setState()`, `InheritedWidget`, `Provider`, `Riverpod`, `BLoC`, and more.

23. **What are `InheritedWidget` and `InheritedModel` in Flutter?**

    - `InheritedWidget` is used to propagate data down the widget tree to descendant widgets. `InheritedModel` is a more advanced version, used when you need to optimize updates based on partial data changes.

24. **What is the role of `GlobalKey` in Flutter?**

    - `GlobalKey` allows access to the state of a `StatefulWidget` across the widget tree, even from different parts of the app.

25. **What is the `Flutter Inspector` tool?**

  - The Flutter Inspector is a tool in Android Studio and Visual Studio Code that helps in debugging the widget tree, viewing widget properties, and inspecting layouts in a Flutter application.

Intermediate Questions (26-50)

26. **What is a `CustomPainter` in Flutter?**

  - `CustomPainter` is a widget that allows drawing custom graphics onto the canvas, often used for advanced drawing or animations.

27. **What are `FutureBuilder` and `StreamBuilder` in Flutter?**

  - `FutureBuilder` and `StreamBuilder` are widgets that help in building widgets based on asynchronous data, either from a `Future` or a `Stream`.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

28. **What is `Flutter's Hot Restart`?**

  - Hot restart is similar to hot reload but restarts the entire app, resetting the app's state while keeping the code changes.

29. **What is the purpose of `Flutter's` WidgetsBinding` class?**

  - `WidgetsBinding` is responsible for interacting with the Flutter framework and scheduling the rendering of frames, handling input events, etc.

30. **How do you manage routes in Flutter?**

  - Routes can be managed using `Navigator`, `MaterialPageRoute`, or more advanced route management solutions like `FlutterNavigator`, `GetX`, or `AutoRoute`.

31. **How do you handle errors in Flutter?**

- Errors in Flutter can be handled by using `try-catch` blocks, `ErrorWidget.builder`, or Flutter's error handling callback `FlutterError.onError`.

32. **What are the advantages of using `Provider` in Flutter?**

    - `Provider` is a simple and efficient state management solution that uses an `InheritedWidget` to propagate state through the widget tree and notify listeners of state changes.

33. **How can you optimize the performance of a Flutter app?**

    - Performance can be optimized by reducing the widget tree depth, avoiding unnecessary rebuilds, using `const` constructors, efficient image caching, and leveraging `ListView.builder` for large lists.

34. **What is the `GestureDetector` widget used for?**

DM TECHPARK

    - `GestureDetector` is used for detecting user gestures like taps, swipes, and drags on the screen, and performing actions based on them.

35. **What is the `Drawer` widget in Flutter?**

    - The `Drawer` widget is used to create a sliding panel that provides navigation options, typically used for side menus.

36. **What are the differences between `Navigator.push()` and `Navigator.pushReplacement()`?**

    - `Navigator.push()` adds a new route to the stack, while `Navigator.pushReplacement()` replaces the current route with a new one.

37. **How do you handle long-running tasks in Flutter?**

- Long-running tasks can be handled using `Future`, `Stream`, `Isolates`, or background processing packages like `flutter_background_fetch`.

38. **What is a `Stream` in Flutter?**

    - A `Stream` is an asynchronous sequence of data events, often used for continuous data, such as listening for user input or receiving real-time updates.

39. **How do you add dependencies in Flutter?**

    - Dependencies in Flutter are added through the `pubspec.yaml` file, under the `dependencies` section, and then fetched using the `flutter pub get` command.

40. **What is the `Flutter DevTools` suite?**

    - `Flutter DevTools` is a suite of tools for performance profiling, debugging, and inspecting a running Flutter application.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

41. **How do you handle localization in Flutter?**

    - Localization in Flutter is handled using the `intl` package, which provides tools for working with different languages and regional settings.

42. **What are `Keys` in Flutter?**

    - `Keys` are used to preserve the state of widgets in Flutter when they move in the widget tree, typically used in scenarios like lists and animations.

43. **What is the difference between `ListView.builder` and `ListView` in Flutter?**

    - `ListView.builder` is used for dynamically generating list items, which is more efficient than `ListView` when the list is large because it only builds visible items.

44.

What is `flutter_bloc`? - `flutter_bloc` is a popular state management library that implements the BLoC (Business Logic Component) pattern to separate business logic from UI code.

45. **What is the `AspectRatio` widget in Flutter?**
    - `AspectRatio` ensures that a widget has a specific aspect ratio, maintaining the width-to-height ratio.
46. **What is `ListView.separated` in Flutter?**
    - `ListView.separated` creates a list of items with separators between each item, often used for displaying list items with visual dividers.
47. **What is `SizedBox` used for in Flutter?**
    - `SizedBox` is used to give widgets fixed width and height or to create spacing between widgets.
48. **How do you implement animations in Flutter?**
    - Animations in Flutter can be implemented using the `AnimationController`, `Tween`, and `AnimatedBuilder` widgets to create complex animations.
49. **What is the `Cupertino` library in Flutter?**
    - `Cupertino` is a set of iOS-styled widgets available in Flutter, allowing developers to create apps that follow Apple's Human Interface Guidelines.
50. **What is a `FutureBuilder` in Flutter?**

DM TECHPARK

- `FutureBuilder` is a widget that builds itself based on the latest snapshot of a `Future`, providing different UI states like loading, error, and success.

Advanced Questions (51-75)

51. **What is the `InheritedWidget` and when should it be used?**

    - `InheritedWidget` is used to efficiently propagate data down the widget tree, allowing descendant widgets to access shared data.

52. **What are `Streams` and `StreamControllers` in Flutter?**

    - A `Stream` provides asynchronous data, while `StreamController` is used to create a stream that can be used to add and listen to events.

DM TECHPARK GUIDE FOR PERFECT CAREER PATHWAY

53. **How do you manage large-scale app state with Flutter?**

    - Large-scale state management in Flutter is typically handled using advanced state management solutions like `BLoC`, `Redux`, `Riverpod`, or `Provider`.

54. **What is `flutter_hooks`?**

    - `flutter_hooks` is a package that adds hooks, like React hooks, to Flutter for more concise and declarative state management.

55. **What is a `CustomScrollView` in Flutter?**

    - `CustomScrollView` is a widget that allows for creating complex scrollable layouts with different types of slivers (custom scrollable areas).

56. **How do you perform unit testing in Flutter?**

- Unit testing in Flutter can be done using the `test` package to write test cases that ensure the correctness of individual functions or classes.

57. **How can you optimize Flutter app startup time?**

    - You can optimize app startup time by reducing unnecessary imports, using deferred loading for non-critical resources, and ensuring only essential code is executed during startup.

58. **What is `AnimatedContainer` in Flutter?**

    - `AnimatedContainer` allows you to animate changes in a container's properties like size, padding, decoration, and color over time.

59. **What is a `StreamTransformer` in Dart?**

    - A `StreamTransformer` allows you to modify events coming through a stream before passing them to the listener.

IDM TECHPARK
GUIDE'S FOR PERFECT CAREER PATHWAY

60. **What is `flutter_driver` used for?**

    - `flutter_driver` is used for integration testing of Flutter apps, allowing you to test the app's UI and interaction with real devices or simulators.

61. **What is the difference between `Stream` and `Future`?**

    - `Stream` provides multiple asynchronous events, while `Future` handles a single asynchronous event or value.

62. **How does Flutter handle platform-specific code?**

    - Flutter uses platform channels to communicate with native code (Java, Swift, etc.) for functionality that cannot be handled in Dart.

63. **How do you implement dark mode in a Flutter app?**

- Dark mode can be implemented by defining themes in `ThemeData` and switching between the light and dark themes based on system settings or user preference.

64. **What is `flutter_bloc` used for in Flutter?**

    - `flutter_bloc` is a state management library that separates business logic from UI using the BLoC pattern, promoting a reactive programming model.

65. **How do you handle deep linking in Flutter?**

    - Deep linking in Flutter is handled using plugins like `uni_links` to open the app with a specific URL and navigate to the corresponding screen.

66. **What is `flutter_secure_storage` used for?**

    - `flutter_secure_storage` is used for storing sensitive information securely on the device, such as passwords or tokens.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

67. **What are `Futures` and `Async/Await` in Dart?**

    - `Futures` represent asynchronous tasks, and `async/await` is used to handle asynchronous code in a more readable and synchronous-like manner.

68. **What is `flutter_riverpod`?**

    - `flutter_riverpod` is a state management solution that provides a more flexible and scalable approach than `Provider`, supporting both synchronous and asynchronous operations.

69. **How does Flutter support localization?**

    - Flutter supports localization by using the `intl` package, which allows developers to translate text and handle different languages and regions.

70. **What is a `Sliver` in Flutter?**

   ○ A `Sliver` is a portion of a scrollable area that can change dynamically during scrolling, such as `SliverList`, `SliverGrid`, and `SliverAppBar`.

71. **How do you integrate Firebase into a Flutter app?**

   ○ Firebase can be integrated into Flutter apps by adding the necessary Firebase dependencies in `pubspec.yaml` and configuring Firebase in both Android and iOS projects.

72. **What is `Flutter Web`?**

   ○ Flutter Web is an extension of Flutter for building applications that run in the browser, allowing developers to build web applications using the same Flutter codebase.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

73. **What is the `Flutter engine`?**

   ○ The Flutter engine is responsible for rendering, running Dart code, and handling platform channels for native functionality.

74. **What is `async` programming in Dart?**

   ○ `async` programming allows for non-blocking asynchronous code execution, using `Future` and `Stream` to perform tasks like I/O operations without freezing the app.

75. **How do you implement push notifications in Flutter?**

   ○ Push notifications can be implemented using the `firebase_messaging` package, which integrates Firebase Cloud Messaging (FCM) with a Flutter app.

Technical Questions (76-100)

76. **What is the `flutter_native_splash` package used for?**

- `flutter_native_splash` is used to configure the launch screen of a Flutter application, enabling customized splash screens for both Android and iOS platforms.

77. **How do you use the `Future.delayed()` function in Flutter?**

- `Future.delayed()` allows you to execute code after a specified delay, often used for delaying UI changes or adding animations.

78. **What is `Isolate` in Dart?**

- `Isolate` is a unit of concurrency in Dart. It allows for running code in parallel threads, useful for heavy computations without blocking the main thread.

DM TECHPARK GUIDE'S FOR PERFECT CAREER PATHWAY

79. **How does `flutter_bloc` handle asynchronous actions?**

- In `flutter_bloc`, asynchronous actions are typically handled within the `Bloc` class using `Stream` and `yield` to manage state changes.

80. **How do you implement custom fonts in Flutter?**

- Custom fonts can be added by including font files in the `assets` section of `pubspec.yaml` and referencing them in `TextStyle` properties.

81. **What is the difference between `Flutter` and `React Native`?**

- `Flutter` uses Dart and compiles to native code, while `React Native` uses JavaScript and relies on a bridge to communicate with native code.

82. **How do you optimize image loading in Flutter?**

- Image optimization can be done by using caching techniques, such as the `cached_network_image` package, and optimizing image size before displaying.

83. **What is a `TextEditingController` in Flutter?**

   - A `TextEditingController` is used to control the text input in a `TextField` widget, allowing programmatic access to the text and managing cursor positions.

84. **How do you implement pagination in a Flutter app?**

   - Pagination can be implemented by using `ListView.builder` combined with `ScrollController` to detect when the user has scrolled to the bottom and fetch more data.

85. **What is `Flutter Inspector` and how does it help in debugging?**

   - `Flutter Inspector` is a tool that helps visualize the widget tree, check widget properties, and debug layouts in real-time.

DM TECHPARK
GUIDE 2 YOUR PERFECT CAREER PATHWAY

86. **What are `Tear-Off Functions` in Dart?**

   - `Tear-off functions` allow you to pass a function reference instead of invoking a function immediately, typically used with methods or class constructors.

87. **How do you perform integration testing in Flutter?**

   - Integration testing in Flutter is done using the `flutter_driver` package, which allows you to interact with the app and verify the app's UI and behavior.

88. **What is `AutomaticKeepAliveClientMixin` used for in Flutter?**

- `AutomaticKeepAliveClientMixin` is used to keep a widget's state alive when it goes off-screen in a `ListView` or other scrollable widgets.

89. **How do you handle background tasks in Flutter?**

- Background tasks in Flutter can be handled using plugins like `flutter_background_fetch` or `workmanager` to perform tasks while the app is not in the foreground.

90. **What is `flutter_test` used for?**

- `flutter_test` is a package used for writing unit, widget, and integration tests in Flutter, ensuring the app's functionality remains correct over time.

91. **What is the `flutter_launcher_icons` package?**

- `flutter_launcher_icons` is used to generate launcher icons for Android and iOS with different sizes for various screen densities.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

92. **What is the purpose of `SplashScreen` in Flutter?**

- A splash screen is the initial screen that appears when the app is launched, often used for loading resources or displaying branding information.

93. **How do you create a custom widget in Flutter?**

- Custom widgets are created by extending `StatelessWidget` or `StatefulWidget` and defining the layout in the `build()` method.

94. **How do you test if a widget is visible in Flutter?**

- Widget visibility can be tested by using `findsOneWidget`, `findsWidgets`, or similar matcher functions in the `flutter_test` package to assert that a widget appears in the widget tree.

95. **What are `Mixins` in Dart?**

    ○ Mixins are a way to add the functionality of a class to another class without inheritance. They are used to avoid duplication of code.

96. **What is `StreamBuilder` and how does it work?**

    ○ `StreamBuilder` listens to a stream and rebuilds the widget tree whenever a new event is emitted, providing real-time updates for asynchronous data.

97. **What is a `BuildContext` in Flutter?**

    ○ `BuildContext` represents the location of a widget within the widget tree and is used to interact with the widget's parent or ancestor widgets.

IDM TECHPARK
GUIDE`S FOR PERFECT CAREER PATHWAY

98. **How does `Flutter` handle animations?**

    ○ Flutter provides several ways to handle animations, such as using `AnimationController`, `Tween`, and `AnimatedWidget`, or using premade widgets like `AnimatedContainer`.

99. **What are `async/await` in Dart?**

    ○ `async` and `await` allow for asynchronous code to be written in a more readable, synchronous-like way, enabling easy handling of tasks like network requests.

100. **How do you set up Firebase Authentication in Flutter?** - Firebase Authentication can be set up by integrating the `firebase_auth` package, configuring Firebase in both Android and iOS projects, and using the authentication methods like signInWithEmailAndPassword.