**Q-1: - Explain the benefits of using Flutter over other cross-platform frameworks.**

 **Single Code for All Devices: Write your app once and it works on Android, iOS, web, and desktop — no need to maintain separate codebases.**

** Rapid Development: With Hot Reload, you can instantly see updates in your app without restarting, speeding up testing and iteration.**

** Stunning User Interfaces: Flutter comes with a rich set of ready-made widgets, making it easy to design attractive, modern, and responsive UIs.**

** High Speed & Performance: Flutter apps are compiled to native machine code, giving smooth animations and fast execution like a native app.**

** Full Control Over Design: You can build your own custom widgets and animations, allowing total flexibility in app appearance and behaviour.**

** Strong Support & Community: Developed by Google, Flutter has a large, active community, plenty of tutorials, and helpful resources.**

** Consistent Look Across Devices: Your app will look the same on every platform, reducing design inconsistencies and testing efforts.**

** Reduced Development Cost: Maintaining a single codebase saves time, effort, and money compared to developing separate apps for each platform.**

** Easy Integration: Flutter works well with existing code, APIs, and third-party services, making it easier to expand or enhance apps.**

** Growing Ecosystem: A wide range of plugins and packages are available to speed up common tasks like database handling, maps, or authentication.**

.

**Q:2: - Describe the role of Dart in Flutter. What are its advantages for mobile development?**

* **Dart: -**
* **UI Implementation: Dart is used to define, customize, and structure the user interface in Flutter apps, enabling developers to create complex and responsive layouts and components efficiently.**
* **​ Business Logic & State Management: Application flows and interactions are handled entirely in Dart, allowing for a coherent and maintainable codebase.**
* **​ Reactive Programming: Dart’s language features support Flutter’s reactive UI paradigm, ensuring that the interface updates automatically as the underlying state or data changes. ​**
* **Dart good for mobile development: -**
* **Hot Reload: Developers can instantly see code changes reflected in running apps without needing to restart, dramatically increasing productivity. ​**
* **Strong Typing and Error Checking: Dart’s static type system helps catch errors early in the development process, leading to more robust applications.**
* **​ Performance: Dart supports both Ahead-of-Time (AOT) compilation for fast, native-like app performance in production and Just-in-Time (JIT) compilation for a fast development experience. ​**
* **Cross-Platform Code: A single Dart codebase can be compiled to run natively on multiple platforms, including iOS, Android, web, and desktop, reducing development effort and improving maintainability. ​**
* **Productivity: Simple and clean syntax lowers the learning curve, while powerful tooling and a rich package ecosystem make Dart a highly productive language for both frontend and backend development.**

**Q-3: - Outline the steps to set up a Flutter development environment**

1. **Install Flutter SDK** – **Download the Flutter SDK from the official website and extract it to a desired location**.
2. **Update Path Variable** – **Add the Flutter bin directory to your system’s PATH**.
3. **Install IDE** – **Install an IDE such as** **Android Studio**, **VS Code**, **or** **IntelliJ IDEA**.
4. **Install Flutter & Dart Plugins** – **Add Flutter and Dart plugins in your chosen IDE**.
5. **Install Android Studio Components** – **Install Android SDK, Android Emulator, and relevant platform tools.**
6. **Set Up an Emulator or Device** – **Configure an Android or iOS emulator or connect a physical device.**
7. **Run flutter doctor** – **Open terminal and run flutter doctor to check for missing dependencies and follow suggested fixes**.
8. **Create a New Flutter Project** **– Use IDE or command line (flutter create project name) to start a project.**
9. **Run the App** – **Launch the app on an emulator or device to verify the setup.**

**Q-4: - Describe the basic Flutter app structure, explaining main. dart, the main function, and the widget tree.**

* **main. dart: This is the main file of your app. It’s where your app starts.**
* **main () Function:**
* **It’s the starting point of every Dart program.**
* **Example:**
* **void main () {**
* **runApp (MyApp ());**
* **}**
* **Widget :**
* **Everything in Flutter is a widget (buttons, text, layouts, etc.).**
* **Widgets are arranged in a tree (some widgets contain other widgets).**
* **Example:**
* **class MyApp extends Stateless Widget {**
* **@override**
* **Widget build (Build Context context) {**
* **return Materia Lapp**
* **(home: Scaffold (**
* **appBar: AppBar (title: Text('Hello')),**
* **body: Centre (child: Text ('Welcome to Flutter')),**