



## Flutter Course Details

Address : Shreeram society, lane no.3 khandagle complex, kharadi bypass pune-411014

---

### Module 1: Introduction to Flutter

- **Introduction to Cross-Platform Development**
  - Understanding the need for cross-platform development
  - Overview of popular frameworks
- **Introduction to Flutter**
  - What is Flutter?
  - Benefits of using Flutter
  - Comparison with other frameworks
- **Setting Up Flutter Development Environment**
  - Installing Flutter SDK
  - Setting up IDE (Android Studio, Visual Studio Code)
  - Running the first Flutter app

### Module 2: Dart Programming Language

- **Introduction to Dart**
  - Overview of Dart language
  - Dart SDK and tools
- **Dart Basics**
  - Variables and Data Types
  - Control Structures (if-else, switch-case, loops)
  - Functions and Methods
- **Object-Oriented Programming in Dart**
  - Classes and Objects
  - Inheritance and Polymorphism
  - Interfaces and Abstract Classes
- **Advanced Dart**
  - Exception Handling
  - Asynchronous Programming (Futures, Streams)
  - Generics and Collections

### Module 3: Flutter Basics

- **Flutter Architecture**
  - Understanding the Flutter framework
  - Flutter's widget tree
- **Introduction to Widgets**

- Stateless vs Stateful Widgets
- Basic Widgets (Text, Image, Icon, Button)
- **Building Your First Flutter App**
  - Project structure
  - Creating a simple UI
  - Hot Reload and Hot Restart

## **Module 4: Layouts and Widgets**

- **Basic Layouts**
  - Containers, Rows, Columns
  - Padding, Margins, and Alignment
- **Advanced Layouts**
  - Stacks and Positioned
  - GridView and ListView
  - Custom Layouts
- **Input and Forms**
  - TextField and Form Widgets
  - Validating input
  - Handling form submission

## **Module 5: State Management**

- **Understanding State Management**
  - Why state management is important
  - State management options in Flutter
- **Basic State Management**
  - Using setState
  - Lifting state up
- **Advanced State Management**
  - Provider package
  - Riverpod, Bloc, and other state management solutions

## **Module 6: Navigation and Routing**

- **Basic Navigation**
  - Navigator and Routes
  - Push and Pop methods
- **Named Routes**
  - Defining named routes
  - Navigating using named routes
- **Passing Data Between Screens**
  - Sending and receiving data
  - Using arguments and return values

## **Module 7: Networking and APIs**

- **Networking Basics**

- Introduction to RESTful APIs
- Making HTTP requests with the http package
- **JSON Parsing**
  - Decoding JSON
  - Working with complex JSON structures
- **Advanced Networking**
  - Interceptors and Network Management
  - Handling errors and exceptions

## **Module 8: Data Persistence**

- **Local Storage**
  - Using Shared Preferences
  - Working with SQLite
- **Cloud Storage**
  - Introduction to Firebase
  - Firestore and Realtime Database
  - Authentication with Firebase Auth

## **Module 9: Advanced Flutter**

- **Animations**
  - Introduction to animations
  - Implicit and Explicit animations
  - Custom animations and transitions
- **Custom Widgets**
  - Building custom widgets
  - Composition and Inheritance
- **Platform-Specific Code**
  - Accessing native features
  - Method channels for platform integration

## **Module 10: Testing and Debugging**

- **Debugging**
  - Debugging tools in Flutter
  - Using the Flutter DevTools
- **Unit Testing**
  - Writing unit tests
  - Mocking dependencies
- **Widget and Integration Testing**
  - Writing widget tests
  - Writing integration tests

## **Module 11: Deployment**

- **Preparing for Release**
  - App versioning and build configurations

- Code obfuscation and minification
- **Publishing to App Stores**
  - Publishing to Google Play Store
  - Publishing to Apple App Store
- **Web and Desktop Deployment**
  - Building for the web
  - Building for desktop (Windows, macOS, Linux)

## **Module 12: Best Practices and Advanced Topics**

- **Code Organization and Architecture**
  - MVC, MVVM, and other architectures
  - Modular coding practices
- **Performance Optimization**
  - Identifying performance bottlenecks
  - Optimizing build methods and rendering
- **Security Considerations**
  - Securing data and APIs
  - Handling sensitive information

## **Conclusion and Final Project**

- **Final Project**
  - Developing a complete Flutter application from scratch
  - Implementing learned concepts
  - Code review and optimization
- **Course Review and Q&A**
  - Recap of key concepts
  - Addressing student questions and feedback
- **Certification**
  - Course completion certificate
  - Guidance on further learning paths and resources