



Course Name: Flutter Professional Certification Course

Course Overview:

This comprehensive curriculum is designed to equip learners with the skills to build advanced mobile applications using Flutter and Dart. Starting with the basics of Flutter widget development, participants will explore cross-platform and native app development, diving deep into Flutter's architecture and the Dart programming language. The course includes hands-on projects such as a timer app, a calculator, and more complex applications like a camera app and a to-do list. Advanced modules cover device hardware integration, API usage, Firebase authentication and storage, and state management using Bloc and Riverpod with Dio. The capstone project involves creating a fully functional app that integrates multiple APIs, authenticates users, and manages data storage. This curriculum is ideal for those aiming to master mobile app development from foundational to advanced levels.

What You'll Learn?

In this comprehensive curriculum, participants will learn how to develop cross-platform mobile apps using Flutter and Dart, focusing on creating robust applications that integrate with various APIs, manage state effectively with advanced techniques like Bloc and Riverpod, and utilize Firebase for authentication and storage solutions.

Duration: 42 hours (7 weeks)

Requirements:

Android Studio

Pre-requisite:

No pre-requisite

Curriculum

Module 1. Dart Flutter Widget Basics

1. Flutter introduction

- Introduction to Cross-platform Development
- Native App Development v/s Cross-platform Development
- Introduction to Flutter
- Features of Flutter
- What is Gradle build system?
- Advantages of Flutter
- Flutter Framework
- Getting Started with Flutter

Project 1 - Timer app

- Creating a blank flutter project and understanding its structure
- Understanding hot reload option
- Understanding basic flutter widgets like row, column, buttons
- "Case Study - Creating a stop watch timer app"

2. Dart fundamentals

- Introduction to Dart
- Data Types in Dart
- Variables in Dart
- Functions in Dart
- Decision Making and Loops
- Object-Oriented Programming
- Continue and Break
- Mixins in Dart
- How super keyword works in Dart
- Inheritance example in flutter

Project 2 - Calculator App

- Taking inputs from users
- How keys can be used to distinguish in form
- Understanding stateful widgets
- Enums in Dart
- Switch cases in Dart
- "Case Study - Creating a simple calculator app"



Project 3 - Page Views in Flutter

- How to create swipeable views
- Creating a simple listview builder
- Creating custom models in Dart
- Adding click event listeners to product items
- "Case Study - Developing a product listing page with custom models"

Module 2. Accessing device hardware in your code

Project 4 - Camera

- How hardware's can be interfaced in flutter
- Asking permissions for accessing camera
- Creating camera app for clicking photos
- Saving files in internal storage
- Recording videos in flutter
- "Case Study - Displaying image and video preview for camera"

Project 5 - To Do app

- Introduction to state management
- Understanding providers in flutter
- What are observers and observables in code
- How consumer widgets work in state management
- "Case Study - Creating a to do app "

Project 6 - Offline Storage app

- How offline storages are important
- Introduction to SQLite
- Creating a simple form with custom validation
- How NoSQL database works in offline storage
- How SOLID principles can shape a better app
- "Case Study - Creating a simple CRUD operation app with NoSQL in flutter"

Project 7 - (Good to have) Using location and google maps in app

- Getting user location from GPS in app
- Understanding permissions for location in app
- How to copy external database to internal storage in app
- Generating a google maps API key
- "Case Study - Creating an app in which you can plot user location on google maps"

Module 3. Bloc State management in flutter

3. Routing

- How simplified routing can be useful to app
- Good practices in routing
- Handling routing for flutter web, mobile and iOS

Project 8 - Immutability in flutter

- What is bloc state management?
- How to create immutable states
- How to have automatic code generation
- Bloc-to-bloc communication
- "Case Study - Creating a simple intro screen using state management"

Module 4. Working with API web services

Project 9 - Integrating APIs in APP

- Introduction to API's
- API calls in flutter
- How JSON serialization and deserialization is done in flutter
- Integrating a GET API call for products
- "Case Study - Designing screen for displaying list of products"
- Creating a product description screen
- How to send data to server using POST API calls

Module 5. Using firebase Authentication and Storage in app

Project 10 - Firebase Authentication

- Introduction to firebase
- Registering app on firebase
- Setting up email providers for gmail authentication
- Creating in gmail sign in feature
- How to create a sign up page
- "Case Study - Signing user to firebase"
- Signing up user to firebase
- Logout user on firebase

Module 6. Using Riverpod state management with Dio

4. Flutter Hooks

- Introduction to hooks
- How hooks can be an improvement against stateless and stateful widget
- How to use your project for flutter web
- Creating a custom callback hook

Project 12 - Dio plugin

- How to use Riverpod state management in app
- Using hooks state management with riverpod
- Using state notifiers in app
- Setting up API calls with Dio plugin
- How to implement app interceptors
- Handling error scenarios with Dio custom exceptions
- "Case Study - Creating a weather forecast analysis project with riverpod and Dio"

Module 7. Using Firebase storage in app

5. Firebase Services

- Introduction to firebase storage
- Using firebase storage for flutter apps
- Accessing firebase storage for images
- Introduction to firebase real time storage
- Saving data to real time storage
- Fetching data from firebase real time DB
- Using firebase remote config in app

6. Capstone Project

Create an app in which you need to integrate API call which will fetch movie details from server. On fetching data display list of movies in a list format. Once a user clicks on a particular movie item then show the movie description in a separate screen. Authenticate user and save their details to firebase. Retrieve latest releases and banner images from firebase storage