6-Day Flutter Course Content

# Day 1: Introduction to Flutter & Setting Up

## Introduction to Flutter

Overview of Flutter, its purpose, how it compares to other frameworks (React Native, Xamarin), Flutter’s architecture (Dart VM, rendering engine). Run a “Hello World” example to introduce widgets.

## Setting Up the Development Environment

Detailed steps for installing the Flutter SDK, setting up Android Studio and Xcode (if using macOS), setting up VS Code as an alternative IDE. Running Flutter Doctor to check setup and troubleshoot any issues.

## Project Structure

Explanation of the lib, test, android, and ios folders. Modify the default main.dart to display a personalized message.

## Dart Basics

Overview of Dart syntax, common data types, control flow (if, for, while), functions. Basic Dart syntax samples and exercises on loops and conditionals.

## Introduction to Widgets

Introduction to Scaffold, AppBar, Container, Text widgets, Row, and Column for basic layout. Build a simple app with text and a basic layout using Rows and Columns.

# Day 2: Working with Widgets and Layouts

## Stateless vs Stateful Widgets

Differences between Stateless and Stateful widgets; when to use each. Create a small app that switches between two widgets to show state handling.

## Basic Widgets in Depth

Text, Icon, Image, and Button widgets with properties and customization. Sample layout including icons and images.

## Layout Widgets

Detailed breakdown of Row, Column, Stack, ListView, GridView. Create a layout with nested widgets, using Row and Column for structure.

## Handling User Input

Widgets like TextField, Switch, Slider, Checkbox, Radio, and how to handle user input. Capture user input using a TextField and display it on screen.

# Day 3: Navigation and State Management

## Navigation Basics

Introduction to routes, Navigator, and how to navigate between screens. Create two screens and navigate back and forth using Navigator.

## Named Routes

Benefits of named routes for larger apps, implementing named routes. Sample app using named routes to navigate between multiple pages.

## Introduction to State Management

setState and Provider basics, how state management works in Flutter. Create a counter app using setState and rebuild it using Provider.

# Day 4: Forms, Data Handling, and Networking

## Forms and Input Validation

Form and TextFormField, validating user input. Create a form with fields like name, email, and phone, validate each.

## Networking Basics

HTTP package, GET and POST requests, JSON parsing. Fetch and parse JSON data from a public API.

## Fetching Data from an API

Displaying data in ListView after fetching it. Integrate an API, display fetched data in a scrollable ListView.

# Day 5: Local Data Storage and Firebase Integration

## Local Data Storage

Using Shared Preferences for lightweight storage. Storing user preferences and retrieving them on app load.

## Firebase Authentication

Setting up Firebase Authentication, sign-in and sign-out methods. Create a sign-in screen using Firebase Authentication.

## Firebase Firestore

CRUD operations using Firestore, real-time updates. Build a to-do list app, storing and retrieving data from Firestore.

# Day 6: Animations, Deployment, and Best Practices

## Basic Animations

Introduction to animations, AnimatedContainer, FadeTransition. Create a simple animation using AnimatedContainer.

## Advanced Animations

Hero animations, creating custom animations. Add Hero animation between screens for smoother transitions.

## App Deployment

Steps for preparing an app for release, building for Google Play and App Store. Walkthrough of building APK and iOS bundles.

## Best Practices

Code structure, performance optimization, responsive design. Examples of responsive design with LayoutBuilder and MediaQuery.