Mobile Application Development with cross-platform

Duration: 12 days

Prerequisites: Knowledge of OOPS-based programming language

Training Outcome

Understand **React & React Native, Dart and Flutter**
✅ Be able to **build UI, handle state, and integrate APIs**
✅ Implement **Firebase Authentication & Firestore Database**
✅ Develop a **fully functional mobile app**
✅ Gain experience with **real-world mobile app development**

🗓️ Day 1: Introduction to Dart Programming

- **Theory:**
    - Overview of Dart and Its Role in Flutter
    - Dart Basics: Variables, Data Types, Operators
    - Control Flow: Loops & Conditional Statements
    - Functions & Object-Oriented Programming (Classes, Objects, Inheritance)
- **Lab:**
    - Install Dart SDK & Run Basic Dart Programs
    - Create Simple Functions & Work with Lists, Maps
    - Implement a Small Console-Based Calculator

🗓️ Day 2: Getting Started with Flutter

- **Theory:**
    - Overview of Flutter & Its Architecture
    - Setting Up Flutter Development Environment
    - Flutter Project Structure & File System
    - Introduction to Widgets: Stateless vs. Stateful Widgets
- **Lab:**
    - Install Flutter SDK & Setup Android/iOS Emulator
    - Create a "Hello World" Flutter App
    - Explore and Modify the Flutter Project Structure

🗓️ Day 3: UI Development & Layouts in Flutter

- **Theory:**
  - Understanding the Widget Tree & Composition
  - Basic UI Components: Text, Images, Buttons
  - Layout Widgets: Column, Row, Stack, ListView
  - Styling, Theming, and Responsive UI Design
- **Lab:**
  - Build a Simple Login Screen UI
  - Create an Interactive Profile Page with Images & Buttons
  - Implement Flexbox-like Layouts using Row & Column

🗓️ Day 4: State Management & Forms

- **Theory:**
  - Understanding State in Flutter
  - Managing State using setState()
  - Handling Forms & User Input
  - Form Validation & Error Handling
- **Lab:**
  - Create a Registration Form with Input Fields
  - Implement Form Validation (e.g., Email, Password)
  - Build a To-Do List App Using Stateful Widgets

🗓️ Day 5: Networking, API Integration & Navigation

- **Theory:**
  - Working with APIs & HTTP Requests (http package)
  - Fetching & Displaying JSON Data
  - Navigation in Flutter (Navigator 1.0 & 2.0)
  - Persistent Data Storage (SharedPreferences & Local Database)
- **Lab:**
  - Fetch & Display Data from a Public API (e.g., Weather API)
  - Implement Multi-Screen Navigation

- Store User Preferences Using SharedPreferences

🗓️ Day 6: End Project – Building a Complete App

- **Project Overview:**
  - Build a **Mini E-commerce App, To-Do List App, or Weather App**
  - Implement **User Authentication (Optional)**
  - Fetch & Display API Data
  - Use **State Management & Navigation**
  - Store **User Preferences Locally**
- **Lab:**
  - Develop the project step by step
  - Debugging & Optimization
  - Final Testing & Deployment Overview

🗓️ Day 7: Introduction to React Basics

- **Theory:**
  - What is React? Understanding Component-Based Architecture
  - JSX Syntax & Rendering Elements
  - Props & State in React
  - Handling Events & Basic Hooks (useState, useEffect)
- **Lab:**
  - Install Node.js, VS Code, and Create a React Project
  - Build a Simple Counter App using useState
  - Implement a Basic To-Do List with State Management

🗓️ Day 8: Getting Started with React Native

- **Theory:**
  - Introduction to React Native & Its Architecture
  - Setting Up React Native Development Environment (Expo & React Native CLI)
  - Core Components: View, Text, Image, Button, ScrollView
  - Styling in React Native (Flexbox, Stylesheets)

- **Lab:**
    - Install Expo CLI & Create a React Native App
    - Build a Simple Profile Screen with Text & Images
    - Implement Flexbox Layouts for Responsive UI

🗓️ **Day 9: Navigation & Multi-Screen Apps**

- **Theory:**
    - Understanding Navigation in Mobile Apps
    - Setting Up React Navigation (react-navigation package)
    - Stack Navigation vs. Tab Navigation
    - Passing Data Between Screens
- **Lab:**
    - Implement Stack Navigation with Multiple Screens
    - Create a Bottom Tab Navigation for an App
    - Pass User Data Between Screens

🗓️ **Day 10: API Integration & Data Fetching**

- **Theory:**
    - Fetching Data from APIs (fetch, axios)
    - Handling API Responses & Errors
    - Displaying Dynamic Data in Components
    - Async/Await & Performance Considerations
- **Lab:**
    - Fetch and Display Data from a Public API (e.g., Weather API)
    - Implement a Search Feature
    - Handle Loading & Error States in API Calls

🗓️ **Day 11: State Management , Authentication & Data Storage**

- **Theory:**
    - Introduction to State Management (Context API, Redux)
    - Managing Global State vs. Local State

    - Persistent Storage using AsyncStorage
    - Debugging & Performance Optimization
- **Lab:**
    - Implement a Global State using Context API
    - Store User Preferences Locally (AsyncStorage)
    - Optimize & Debug State Updates

Firebase Authentication & Database Integration

- **Theory:**
    - Introduction to Firebase
    - Setting Up Firebase Project & Configuring SDK
    - User Authentication (Sign-up, Login, Logout)
    - Firestore Database & CRUD Operations
- **Lab:**
    - Implement Firebase Email/Password Authentication
    - Create a User Profile Screen After Login
    - Store & Retrieve Data in Firestore Database

📅 **Day 12: End Project – Building a Complete App**

- **Project Overview:**
    - Build a **Mini E-commerce App, To-Do List, or News App**
    - Implement **User Authentication (Optional)**
    - Fetch & Display API Data
    - Use **State Management & Navigation**
    - Store **User Preferences Locally**
- **Lab:**
    - Develop the project step by step
    - Debugging & Optimization
    - Final Testing & Deployment Overview