Mobile App Development Roadmap (Beginner → Advanced)

Phase 1: Core Foundations

Before diving into frameworks, build strong basics.

1. **Programming Fundamentals**
   - Variables, data types, loops, functions
   - Object-Oriented Programming (OOP) basics
   - Error handling, debugging
   - Suggested languages: **JavaScript, Dart, Kotlin basics, Swift basics**
2. **Mobile App Concepts**
   - What is Native vs. Cross-platform?
   - Difference between iOS and Android app lifecycle
   - UI/UX basics (wireframes, navigation, layouting)
3. **Tools Setup**
   - Install **Android Studio**, **Xcode (for macOS)**, **VS Code**
   - Learn about **Emulators & Simulators**
   - Git & GitHub basics for version control

Phase 2: Cross-Platform Development

Since you want both Flutter & React Native, start with **one first**, then learn the other.

A. Flutter (Dart)

1. **Dart Language**
   - Variables, collections, async/await
   - Classes, inheritance, mixins
2. **Flutter Basics**
   - Widgets (Stateless vs Stateful)
   - Layouts (Row, Column, Stack, ListView, GridView)
   - Navigation & Routing
   - Forms & Input handling
3. **Intermediate**
   - State management (Provider, Riverpod, Bloc, GetX)
   - API calls (REST, JSON parsing, Dio/http package)
   - Local storage (SharedPreferences, Hive, SQLite)
4. **Advanced**
   - Firebase integration (Auth, Firestore, Push Notifications)
   - Animations & Custom UI
   - Publishing apps (Play Store & App Store)

B. React Native (JavaScript/TypeScript)

1. **JavaScript Refresher**
   - ES6+, async/await, arrow functions
   - React basics (components, hooks, props, state)
2. **React Native Basics**
   - Core components (View, Text, Image, ScrollView)
   - Flexbox layout
   - Navigation (React Navigation)
3. **Intermediate**
   - API integration (Axios/fetch)
   - State management (Context, Redux, Zustand, Recoil)
   - Local database (AsyncStorage, SQLite, Realm)
4. **Advanced**
   - Firebase integration
   - Push notifications
   - Native modules (bridging with Kotlin/Swift)
   - Publishing apps

Phase 3: Native Development

Once you're comfortable with cross-platform, go deeper into **native languages**.

A. Kotlin (Android Native)

1. **Kotlin Basics**
   - Variables, functions, OOP, coroutines
   - Android Studio setup
2. **Android Development**
   - Activities, Fragments, Lifecycle
   - UI with XML & Jetpack Compose (modern way)
   - RecyclerView, Navigation Component
   - Data persistence (Room, SharedPreferences)
3. **Advanced**
   - REST API with Retrofit
   - Firebase (Push, Auth)
   - Background services & WorkManager
   - Publishing Android apps

B. Swift (iOS Native)

1. **Swift Basics**
   - Variables, functions, structs, classes

    - Optionals, error handling
    - Protocols, extensions
2. **iOS Development**
    - Xcode interface
    - UIKit basics (Storyboard, AutoLayout)
    - SwiftUI basics (modern UI toolkit)
    - Navigation & Lists
3. **Advanced**
    - API integration with URLSession/Alamofire
    - CoreData & UserDefaults
    - Notifications, Background tasks
    - App Store publishing

Phase 4: Advanced & Professional Development

- **Architecture patterns**: MVVM, Clean Architecture
- **Testing**: Unit testing, Integration testing
- **CI/CD**: Fastlane, GitHub Actions
- **App Security**: Secure storage, API security
- **Performance Optimization**

Phase 5: Build Projects

To solidify skills, build **real-world projects**:

1. **Beginner Projects**
    - ToDo App
    - Weather App
    - Notes App
2. **Intermediate**
    - Chat App (Firebase)
    - E-commerce App (Cart, API, Auth)
    - News App (REST API, caching, offline support)
3. **Advanced**
    - Food Delivery App (cart, payment, maps, notifications)
    - Social Media App (posts, likes, comments, realtime)
    - Finance Tracker (charts, transactions, cloud sync)

Phase 6: Mastery & Career

- Contribute to **open-source projects**
- Build a **portfolio with GitHub & Play Store/App Store apps**

- Learn **Monorepos & Micro-frontend apps**
- Explore **Cross-platform with native integrations (e.g., Flutter + native Swift/Kotlin plugins)**