# React Native Interview Q&As (Part 3 Advanced Topics & Deployment)

## 🔹 Animations & UI Enhancements

**Q1. What is the Animated API in React Native?**
Answer: Provides declarative animations for UI elements (opacity, translate, scale). Works with Animated.Value and Animated.timing.
💡 Tip: Use for smooth, customizable animations.

**Q2. What is LayoutAnimation?**
Answer: Enables automatic animation of layout changes (e.g., expanding/collapsing views).
💡 Tip: Great for list or UI transitions.

**Q3. What is React Native Reanimated?**
Answer: High-performance animation library that runs animations on the native thread.
💡 Tip: Preferred for complex animations.

**Q4. What is Gesture Handler in React Native?**
Answer: Handles advanced gestures (swipe, drag, pinch, pan) with better performance than default touchables.
💡 Tip: Combine with Reanimated for interactive gestures.

## 🔹 Native Modules & Platform APIs

**Q5. What are Native Modules in React Native?**
Answer: Allow JS code to call platform-specific code (Java/Kotlin for Android, Swift/Obj-C for iOS).
💡 Tip: Use when device features aren't available in JS.

**Q6. How do you create a custom Native Module?**
Answer: Write native code in Java/Swift, expose it via a bridge, and

use it in JS.
💡 Tip: Know the steps for interviews, even if rarely implemented.

**Q7. What is the use of Platform module?**
Answer: Detects OS (Platform.OS === "android" | "ios") for conditional rendering.
💡 Tip: Use for platform-specific UI or logic.

**Q8. How do you integrate third-party native libraries?**
Answer: Install via npm/yarn, then link (auto-linking in RN 0.60+).
💡 Tip: Mention manual linking for older RN versions.

## 🔹 Background Tasks & Notifications

**Q9. How to implement push notifications in React Native?**
Answer: Use Firebase Cloud Messaging (Android) or APNs (iOS). Libraries like react-native-push-notification help.
💡 Tip: Always mention platform differences.

**Q10. What is Background Fetch in React Native?**
Answer: Runs periodic tasks (like syncing data) when app is in background.
💡 Tip: Use react-native-background-fetch.

**Q11. Can React Native apps run background location tracking?**
Answer: Yes, using Geolocation APIs + background permissions.
💡 Tip: Requires extra setup on iOS (Always Allow).

**Q12. How do you handle permissions for background tasks?**
Answer: Use PermissionsAndroid (Android) and Info.plist settings (iOS).
💡 Tip: Always request runtime permissions.

## 🔹 Deployment & CI/CD

**Q13. Difference between Debug and Release builds?**
Answer: Debug → Includes dev tools, slower. Release → Optimized, signed, and used for production.
💡 Tip: Always build in release mode before publishing.

**Q14. How do you sign an Android app for release?**
Answer: Generate Keystore, add signing configs in gradle.properties, build APK/AAB.
💡 Tip: Don't lose your keystore!

**Q15. How do you sign an iOS app for release?**
Answer: Use Apple Developer account, provisioning profiles, and certificates.
💡 Tip: Xcode manages signing, but you must enroll in Apple Dev Program.

**Q16. How do you publish a React Native app to Play Store?**
Answer: Generate AAB, upload to Play Console, provide app details, and submit.
💡 Tip: AAB is now mandatory, not APK.

**Q17. How do you publish a React Native app to App Store?**
Answer: Archive build in Xcode → Upload via Transporter → Submit in App Store Connect.
💡 Tip: Apple review process takes longer than Google.

**Q18. What is the role of Fastlane in React Native?**
Answer: Automates builds, signing, and deployment for Android/iOS.
💡 Tip: Mention CI/CD pipelines.

**Q19. What CI/CD tools are used with React Native?**
Answer: GitHub Actions, Bitrise, CircleCI, App Center.
💡 Tip: Highlight automation for testing & publishing.

**Q20. How do you reduce app size in React Native?**
Answer: Enable Proguard (Android), remove unused assets, use Hermes engine.
💡 Tip: Keep bundle size small for performance.