React Native Interview Questions – Part 2 (Advanced & Practical)

🔷 Section 1 Navigation & Performance

Q1. What is React Navigation?
Answer: Popular library for routing & navigation in React Native apps. Supports stack, tab, drawer navigation.
💡 Tip: Mention deep linking & dynamic routing.

Q2. Difference between Stack Navigator and Tab Navigator?
Answer: Stack → Screens in stack order (push/pop). Tab → Bottom/Top navigation bar with multiple screens.
💡 Tip: Use Stack for flows, Tab for main sections.

Q3. How do you optimize FlatList performance?
Answer: Use keyExtractor, getItemLayout, initialNumToRender, removeClippedSubviews.
💡 Tip: Always use FlatList instead of ScrollView for large lists.

Q4. How to handle large images in React Native?
Answer: Use Image with resizeMode, caching libraries (FastImage).
💡 Tip: Optimize image size before bundling.

Q5. What is VirtualizedList?
Answer: Base component for FlatList & SectionList, optimizes rendering for large datasets.
💡 Tip: FlatList is wrapper of VirtualizedList.

🔷 Section 2 Styling & UI

Q6. How do you style components in React Native?
Answer: Inline styles, StyleSheet.create, or UI libraries (NativeBase, React Native Paper).
💡 Tip: Use StyleSheet for performance (static object).

Q7. What is the difference between Flexbox in React Native vs Web?
Answer: RN uses Flexbox by default with flexDirection: 'column' (web → row).
💡 Tip: Always remember RN default direction is **column**.

Q8. How to make responsive UI in React Native?
Answer: Use Dimensions, PixelRatio, Flexbox, react-native-responsive-screen.
💡 Tip: Avoid fixed px values.

Q9. What is SafeAreaView?
Answer: Wrapper that respects notches, status bars, safe areas on iOS/Android.
💡 Tip: Always wrap root container with SafeAreaView.

Q10. Difference between TouchableOpacity & Pressable?
Answer: TouchableOpacity → simple button with opacity effect. Pressable → newer, more flexible touch handling.
💡 Tip: Prefer Pressable for modern apps.

🔹 Section 3 State Management & Data

Q11. How do you manage global state in React Native?
Answer: Redux, MobX, Zustand, or Context API.
💡 Tip: Use Redux Toolkit for cleaner code.

Q12. Difference between Redux and Context API?
Answer: Redux → powerful for large apps, middleware support. Context → simple global state for small apps.
💡 Tip: Interviewers like Redux Toolkit mention.

Q13. What is AsyncStorage?
Answer: Key-value storage system for persisting small data (token, preferences).
💡 Tip: Don’t use for large/secure data.

Q14. How to store secure data in React Native?
Answer: Use react-native-keychain or encrypted storage.
💡 Tip: Never store sensitive data in AsyncStorage.

Q15. How do you connect React Native app with APIs?
Answer: Using Fetch API or Axios for HTTP requests.
💡 Tip: Always handle errors & loading states.

🔹 Section 4 Native Modules & Integrations

Q16. What are Native Modules?
Answer: Custom code in Java/Kotlin (Android) or Swift/Obj-C (iOS) exposed to JS.
💡 Tip: Useful for device features unavailable in JS.

Q17. How to access device camera in React Native?
Answer: Use react-native-camera or expo-camera.
💡 Tip: Remember permissions handling.

Q18. How to send push notifications in React Native?
Answer: Use FCM (Firebase Cloud Messaging) or OneSignal.
💡 Tip: Mention both Android & iOS setup.

Q19. How do you handle deep linking?
Answer: Configure with React Navigation linking or Branch.io.
💡 Tip: Used for opening app via URL/scheme.

Q20. What is the Linking API?
Answer: Used to open URLs, dialers, email apps from RN.
💡 Tip: Always check with canOpenURL.

🔹 Section 5 Debugging & Testing

Q21. How do you debug React Native apps?
Answer: Flipper, React DevTools, Console.log, Remote JS Debugging.
💡 Tip: Flipper is preferred in interviews.

Q22. What are common performance bottlenecks in React Native?
Answer: Overuse of re-renders, heavy animations, large lists.
💡 Tip: Optimize with memoization, FlatList, native animations.

Q23. How to reduce app bundle size?
Answer: Optimize images, use Hermes engine, enable ProGuard, code splitting.
💡 Tip: Mention Hermes → better startup.

Q24. What is the role of Hermes in React Native?
Answer: Lightweight JS engine for faster startup, reduced memory usage.
💡 Tip: Default for new RN apps.

Q25. How do you test React Native apps?
Answer: Unit testing → Jest. Integration/UI → Detox, Appium.
💡 Tip: Mention Jest snapshot testing.

🔹 Section 6 Deployment & Misc

Q26. How do you deploy a React Native app?
Answer: iOS → App Store via Xcode. Android → Play Store via APK/AAB.
💡 Tip: Mention signing & store guidelines.

Q27. What is Expo?

Answer: Framework with RN for faster dev, provides APIs without native code.

💡 Tip: Easy for beginners, but limited native flexibility.

Q28. Difference between Expo & React Native CLI?

Answer: Expo → easy setup, limited native control. CLI → full native access, harder setup.

💡 Tip: CLI is production choice.

Q29. How do you handle app versioning?

Answer: Managed via app.json (Expo) or build.gradle/Xcode settings.

💡 Tip: Mention semantic versioning.

Q30. How do you handle background tasks in React Native?

Answer: Use react-native-background-fetch or native code.

💡 Tip: Mention OS limitations (battery optimizations).