

React Native Mobile App Development

Build, test and ship Android and iOS apps with React Native, Expo and TypeScript

Mobile App Development · intermediate · Live online, instructor-led

This syllabus is a draft awaiting instructor review and may change.

A fourteen-week live course in React Native with Expo and TypeScript: hooks, Expo Router, Zustand and TanStack Query, APIs and auth, offline storage, native modules, push notifications, testing, EAS Build and store releases.

Length and schedule

84 hours of live teaching, on either schedule:

- Weekday Intensive (Mon–Fri): about 17 weeks before holidays
- Weekend Evening (Sat & Sun evenings): about 21 weeks before holidays

Fees per student, whole course

- Group: Rs. 13,999
- Duo: Rs. 22,999
- One-to-One: Rs. 41,999

Launch fee — GST treatment to be confirmed.

What you will be able to do

- Write type-safe modern JavaScript and TypeScript for React Native
- Build interfaces from React components and hooks with core components and responsive styling
- Structure an app with Expo Router, Zustand for client state and TanStack Query for server state
- Integrate REST APIs with validation, authentication and secure token storage
- Keep the app usable offline with AsyncStorage, SQLite and background sync
- Use native modules, push notifications and a test-mode payment flow
- Test with Jest and React Native Testing Library, automate journeys with Maestro, and profile performance
- Ship signed production builds through EAS Build and EAS Submit to the Play Store and App Store

Before you start

- Working knowledge of JavaScript basics (variables, functions, arrays, objects); TypeScript is taught from the start
- Basic understanding of HTML/CSS concepts and of how apps talk to servers over HTTP with JSON
- Comfortable with Git basics (clone, commit, push) and the command line
- A laptop with 16 GB RAM recommended (8 GB minimum) and around 30 GB free disk space for Node.js, Android Studio and an emulator
- An Android or iOS phone is helpful for device testing but not required
- The iOS Simulator and local iOS builds need a Mac with Xcode; without a Mac you can build iOS apps with the EAS Build cloud service and test on a physical iPhone, or complete the course on Android
- Publishing to the stores needs your own Google Play developer account and Apple Developer Program membership; both are optional, and store steps are demonstrated live

Tools

Node.js and npm, TypeScript, Expo SDK and Expo CLI, Expo development builds, VS Code, Android Emulator and Xcode Simulator, Expo Router, React Navigation, Zustand, Redux Toolkit, TanStack Query, Zod, AsyncStorage, expo-sqlite and expo-secure-store, expo-notifications and Firebase Cloud Messaging, Razorpay React Native SDK (test mode), Jest and React Native Testing Library, Maestro, React Native DevTools, EAS Build, EAS Submit and EAS Update, GitHub Actions, Google Play Console and App Store Connect, Git and GitHub

Syllabus

Module 1: JavaScript, TypeScript and React foundations

The modern JavaScript, TypeScript and React that every React Native screen is built on.

1. Modern JavaScript for React Native (2 h)
2. TypeScript essentials (2 h)
3. TypeScript and JavaScript practice and doubt-clearing (2 h)
4. React components, props and state (2 h)
5. Effects, context and custom hooks (2 h)
6. React review and doubt-clearing (2 h)

Module 2: Core components, styling and Expo

Set up an Expo project with development builds and build native interfaces.

7. Expo, development builds and the React Native architecture (2 h)
8. Core components (2 h)
9. Styling and responsive layout (2 h)
10. Lists and performance-friendly rendering (2 h)
11. Forms, keyboard handling and week 4 review (2 h)

Module 3: Navigation with Expo Router

File-based routing, nested navigators and deep links for a multi-screen app.

12. File-based routing with Expo Router (2 h)
13. Dynamic routes, modals, guards and deep links (2 h)
14. Navigation lab and doubt-clearing (2 h)

Module 4: State management and data fetching

Separate client state from server state and keep both predictable.

15. Client state versus server state (2 h)
16. Redux Toolkit compared with Zustand (2 h)
17. Server state with TanStack Query (2 h)
18. Mutations and optimistic updates (2 h)
19. State review and doubt-clearing (2 h)

Module 5: APIs and authentication

Connect the app to a real backend and sign users in securely.

20. REST API integration (2 h)
21. Authentication flows (2 h)
22. Social sign-in and auth-aware navigation (2 h)
23. Mid-course assessment and doubt-clearing (2 h)

Module 6: Local storage and offline-first apps

Store data on the device and keep the app useful without a connection.

- 24. AsyncStorage and secure storage (2 h)
- 25. Local databases with SQLite (2 h)
- 26. Offline-first sync (2 h)
- 27. Offline lab and doubt-clearing (2 h)

Module 7: Native modules, notifications and payments

Reach platform capabilities, send push notifications and integrate payments safely.

- 28. Native modules and config plugins (2 h)
- 29. Device APIs and permissions (2 h)
- 30. Push notifications (2 h)
- 31. In-app payments with Razorpay (test mode) (2 h)
- 32. Integrations review and capstone kick-off (2 h)

Module 8: Testing and performance

Prove the app works with automated tests and keep it fast.

- 33. Unit and component tests (2 h)
- 34. End-to-end tests with Maestro (2 h)
- 35. Performance profiling (2 h)
- 36. Testing clinic and doubt-clearing (2 h)

Module 9: Release engineering and capstone

Build and submit with EAS, automate releases and complete the capstone project.

- 37. EAS Build and production builds (2 h)
- 38. EAS Submit and store deployment (2 h)
- 39. CI/CD and over-the-air updates (2 h)
- 40. Capstone build lab and doubt-clearing (2 h)
- 41. Capstone review (2 h)
- 42. Capstone presentations and final assessment (2 h)

Final project

Build and release-prepare a production-style React Native app. Choose one brief — a clinic appointment booker, a society maintenance tracker or a small-shop order app — or propose your own of similar scope for approval. Your app must:- be written in TypeScript with Expo and a development build, using Expo Router, Zustand for client state and TanStack Query for a provided REST API (or a backend you build);- support sign-up, sign-in, sign-out and token refresh, with tokens kept in expo-secure-store;- cache data in SQLite so the main screens work offline and pending changes sync when the connection returns;- receive a push notification that deep-links to the right screen;- include one native integration (a device module such as camera or location, or a small custom Expo module) and, where it fits the brief, a Razorpay checkout in test mode verified by the server;- have unit and component tests with Jest and React Native Testing Library and at least one Maestro flow, running in CI on every push. Deliverables: the Git repository with a README covering setup and architecture, an EAS production build for Android (App Bundle), the eas.json and CI workflow files, a short performance note showing one problem you found and fixed, a store-listing draft (description, screenshots and privacy details) and a five-minute recorded or live demo. Projects are reviewed live in the final week; iOS builds are optional and can be made with EAS Build without a Mac.

Certificate

- At least 80% attendance
- All assignments submitted
- Projects passed
- Final assessment passed (60%)

A ZORK Academy course completion certificate; paying for the course does not by itself earn it.

Next step

Book a free consultation: <https://zorkacadamy.online/consultation?course=react-native-app-development>

We help you prepare for work; we do not guarantee jobs, placements or salaries.