

Flutter Mobile App Development

Build, test and ship production Android and iOS apps from one Dart codebase

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 Flutter and Dart: widgets and layout, `go_router`, Riverpod, REST APIs, authentication, offline-first storage, native integrations, push notifications, testing, CI/CD 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 idiomatic, null-safe Dart using `async/await`, streams, classes and mixins
- Build responsive, accessible Flutter interfaces from widgets, layouts and themes
- Structure an app feature-first with `go_router` navigation, Riverpod state and repositories
- Integrate REST APIs with `dio` and `json_serializable`, including authentication and token refresh
- Keep the app usable offline with local storage, a Drift database and background sync
- Add native features through platform channels and plugins, push notifications with FCM and a test-mode payment flow
- Cover the app with unit, widget and integration tests and fix performance problems with DevTools
- Automate builds with CI/CD and prepare signed releases for the Play Store and App Store

Before you start

- Working knowledge of at least one programming language (variables, functions, loops, classes)
- Basic understanding of how apps talk to servers over HTTP and what JSON looks like
- Comfortable with Git basics (clone, commit, push)
- A laptop with 16 GB RAM recommended (8 GB minimum) and around 30 GB free disk space for the Flutter SDK, Android Studio and an emulator
- An Android phone is helpful for device testing but not required
- Building and running on iOS needs a Mac with Xcode, or a cloud build service such as Codemagic; Windows and Linux users can complete the course on Android and learn the iOS release steps
- 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

Flutter SDK and Dart, Android Studio or VS Code with the Flutter extension, Android Emulator and Xcode Simulator, Flutter DevTools, `go_router`, Riverpod, `dio`, `json_serializable` and `build_runner`, `shared_preferences` and `flutter_secure_storage`, Drift, Firebase Cloud Messaging, Razorpay Flutter SDK (test mode), mocktail, GitHub Actions, Codemagic, Google Play Console and App Store Connect, Git and GitHub

Syllabus

Module 1: Dart for Flutter developers

The Dart language features Flutter code relies on, from null safety to streams.

1. Dart essentials and tooling (2 h)
2. Null safety in practice (2 h)
3. Classes, mixins and extensions (2 h)
4. Futures, `async/await` and streams (2 h)
5. Week 2 Dart practice and doubt-clearing (2 h)

Module 2: Widgets, layout and theming

Build Flutter interfaces from widgets and make them responsive and accessible.

6. How Flutter renders: widgets and the tree (2 h)
7. Layout fundamentals (2 h)
8. Lists, grids and slivers (2 h)
9. Theming, responsiveness and accessibility (2 h)
10. Forms, input and week 4 review (2 h)

Module 3: Navigation with `go_router`

Declarative routing, nested navigation and deep links for a multi-screen app.

11. Declarative routing with `go_router` (2 h)
12. Nested navigation, guards and deep links (2 h)
13. Navigation lab and doubt-clearing (2 h)

Module 4: State management and architecture

Manage state predictably with Riverpod and organise code feature-first.

14. From `setState` to Riverpod (2 h)
15. Async state with `Notifier` and `AsyncNotifier` (2 h)
16. Bloc compared with Riverpod (2 h)
17. Feature-first architecture and repositories (2 h)
18. Architecture review and doubt-clearing (2 h)

Module 5: REST APIs and authentication

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

19. HTTP with `dio` and `http` (2 h)
20. JSON models with `json_serializable` (2 h)
21. Error handling and resilient networking (2 h)
22. Authentication flows (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. Key-value and secure storage (2 h)
- 25. Local databases with Drift (2 h)
- 26. Offline-first sync (2 h)
- 27. Offline lab and doubt-clearing (2 h)

Module 7: Native features, notifications and payments

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

- 28. Platform channels and native code (2 h)
- 29. Device plugins and permissions (2 h)
- 30. Push notifications with Firebase Cloud Messaging (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 widget tests (2 h)
- 34. Integration tests (2 h)
- 35. Performance profiling with DevTools (2 h)
- 36. Testing clinic and doubt-clearing (2 h)

Module 9: Release engineering and capstone

Automate builds, publish to the stores and complete the capstone project.

- 37. CI/CD with GitHub Actions and Codemagic (2 h)
- 38. Flavours, signing and production builds (2 h)
- 39. Play Store and App Store deployment (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 Flutter 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:- use a feature-first structure with gorouter, Riverpod and repositories in front of a provided REST API (or a backend you build);- support sign-up, sign-in, sign-out and token refresh, with tokens kept in secure storage;- cache data in Drift so the main screens work offline and pending changes sync when the connection returns;- receive a push notification through Firebase Cloud Messaging that deep-links to the right screen;- include one native integration (a platform channel or a device plugin such as camera or location) and, where it fits the brief, a Razorpay checkout in test mode verified by the server;- have unit, widget and at least one integration test, running in a CI workflow on every push. Deliverables: the Git repository with a README covering setup and architecture, a signed Android release build (App Bundle), the CI workflow file, a short DevTools 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 for learners without a Mac or cloud build access.

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://zorkacademy.online/consultation?course=flutter-app-development>

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