Admissions & Support:
support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: FLUT • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/flutter-training

FLUT • CROSS-PLATFORM MOBILE & DESKTOP ENGINEERING | **BEGINNER TO SENIOR FLUTTER & DART 3 ARCHITECT**

100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

Flutter Training

Master cross-platform Flutter 3 & Dart 3 from the 3-Tree Architecture (Widget, Element, RenderObject) and Impeller shader engine to BoxConstraints, Scaffold, Row/Column/Stack, TextEditingController, GlobalKey Forms, ListView.builder, GridView, TabBar, Gestures & Riverpod/BLoC State Management — with 10×50 MCQ Mock Tests per topic and Spoken Voice + MockAttempt AI Interviews at ₹4,999.

Duration: 15–17 Weeks • Live + Hands-On Projects (208 Hrs) • **Modules:** 16 • **Live Topics:** 42 • **Labs & Tasks:** 84 • **Mocks / AI Vivas:** 244 / 244 • **Projects:** 16 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹4,999** + 18% GST (₹900) = **Total Incl. GST: ₹5,899**

Live Batch Tracks (7:00 AM – 8:00 PM IST Flexible Slots): Weekday Live Online Batch (Mon – Fri • 1 Hr/Day • 7:00 AM – 8:00 PM) (Monday to Friday (1 Hour Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 1-Hour Slot Based on Trainer Availability) • 17–19 Weeks Fast-Track) | Weekend Live Online Batch (Sat – Sun • 2 Hrs/Day • 7:00 AM – 8:00 PM) (Saturday & Sunday (2 Hours Live Session / Day) • 07:00 AM – 08:00 PM IST (Students Can Choose Any 2-Hour Slot Based on Trainer Availability) • 19–21 Weeks Weekend Track)

Core Technologies & Engineering Stack: Flutter 3.24+ & Material 3 • Dart 3 (Records, Sealed Classes, Null Safety) • Impeller GPU Rendering Engine • Widget / Element / RenderObject Trees • Slivers, ListView.builder & GridView • Forms, TextEditingController & Gestures • Riverpod, Provider & BLoC State Architecture

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Understand Flutter’s Impeller AOT shader pipeline, Dart JIT Hot Reload vs AOT ARM64 machine code, and the 3-Tree reconciliation engine
2. Master Flutter’s golden layout rule (Constraints go down, Sizes go up, Parent sets position) across Scaffold, Row, Column, Expanded, and Stack
3. Build leak-free forms, lazy-built 10,000-row ListView/GridView feeds, gesture arenas, and reactive unidirectional state with ValueNotifier/Riverpod
4. Clear 200 Timed Mock Exams (10,000 Flutter & Dart MCQs) and 200 Spoken Voice + MockAttempt AI Cross-Platform Technical Interviews
5. Dart 3 null-safety, records, pattern matching, and sealed state classes for crash-free cross-platform code.
6. Scalable feature-first architecture with Riverpod / BLoC state management and pure repository abstractions.
7. Custom responsive layouts, Slivers, implicit/explicit animations, and platform-adaptive Material/Cupertino UX.
8. Dio REST interceptors, token refresh queues, secure storage, local offline cache, and Firebase/push workflows.

Eligibility & Prerequisites: Basic programming familiarity (variables, loops, functions) — Dart 3 sound null safety, async/await, and OOP are taught from scratch • Any Windows, macOS, or Linux laptop with Flutter SDK, VS Code or Android Studio, and Chrome / Android Emulator / iOS Simulator • No prior Flutter industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (16 MODULES • 42 LIVE TOPICS)

MODULE 1 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 1: Flutter Cross-Platform Paradigm, Impeller Engine & SDK Toolchain Setup

Understand why Flutter draws every pixel directly onto a GPU canvas via Impeller (instead of OEM native UI bridges), Dart’s dual JIT/AOT compiler, and reproducible `flutter doctor` + `pubspec.yaml` setup.

Objectives: ✓ Explain how Flutter achieves pixel-identical 60/120fps UI across iOS, Android, Web, macOS, Windows, and Linux • ✓ Contrast Dart VM JIT stateful Hot Reload during development with AOT native ARM64 compilation in release builds • ✓ Configure the Flutter SDK, `flutter doctor -v` diagnostics, semantic versioning in `pubspec.yaml`, and `pubspec.lock` reproducibility

1.1 Flutter Overview — Own-Canvas GPU Rendering (Impeller), Dart JIT/AOT & Multi-Platform Targets [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Overview — Own-Canvas GPU Rendering (Impeller), Dart JIT/AOT & Multi-Platform Targets in real Flutter systems — specifically: Why Flutter...

1.2 Flutter Installation & Toolchain — `flutter doctor`, `pubspec.yaml`, Assets & FVM [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Installation & Toolchain — `flutter doctor`, `pubspec.yaml`, Assets & FVM in real Flutter systems — specifically: Set up the Flutter SDK c...

Module 1 Hands-On Capstone Project (100 Marks): Flutter Environment Readiness Auditor & Pubspec SemVer Resolver — Build a Dart 3 CLI tool that validates toolchain health checks (`Flutter SDK`, `Android Toolchain`, `Xcode/Cocoapods`, `VS Code`), verifies sound null-safety SDK constraints (`>=3.0.0 <4.0.0...

MODULE 2 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 2: First Flutter App Anatomy & The 3-Tree Architecture (`Widget`, `Element`, `RenderObject`)

Dissect `main() -> runApp()`, `MaterialApp(theme: ThemeData(useMaterial3: true))`, and Flutter's core engine architecture: immutable `Widget` blueprints, mutable `Element` mounts, and `RenderObject` painting.

Objectives: ✓ Structure a clean Flutter entry point with `WidgetsFlutterBinding.ensureInitialized()` and Material 3 `ColorScheme.fromSeed` • ✓ Explain why rebuilding thousands of `Widget` instances is cheap when `Element.canUpdate()` reuses `RenderObject`s • ✓ Trace UI Thread (Dart Isolate) vs Raster Thread (GPU Impeller) responsibilities and Platform Channel async messaging

2.1 First Flutter Application — `main()`, `runApp()`, `MaterialApp` & `const` Constructors [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply First Flutter Application — `main()`, `runApp()`, `MaterialApp` & `const` Constructors in real Flutter systems — specifically: Bootstrap a Materia...

2.2 Flutter Architecture — Framework (Dart), Engine (C++/Impeller), Embedder & The 3 Trees [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Architecture — Framework (Dart), Engine (C++/Impeller), Embedder & The 3 Trees in real Flutter systems — specifically: Explore the 3 archi...

Module 2 Hands-On Capstone Project (100 Marks): 3-Tree Reconciliation Simulator (`Widget.canUpdate` & `const` Short-Circuiting) — Implement an educational simulation of Flutter's `Element.updateChild()` reconciliation algorithm demonstrating how matching `runtimeType` + `key` updates an existing `Element` in place whil...

MODULE 3 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 3: Dart 3 Language Mastery & Stateless vs Stateful Widget Lifecycles

Write expressive sound null-safe Dart 3 code (Records, Pattern Matching, Sealed Classes, Isolates, Futures/Streams) and master `StatelessWidget` vs `StatefulWidget` (`initState`, `didUpdateWidget`, `setState`, `dispose`).

Objectives: ✓ Leverage Dart 3 sound null safety (`?`, `??`, `?.`, `late`), Records, and exhaustive `sealed class` pattern matching • ✓ Manage `StatefulWidget` + `State<T>` lifecycles cleanly from `initState()` through `didUpdateWidget()` to `dispose()` • ✓ Offload heavy CPU parsing (> 16ms) from the main UI Isolate using `Isolate.run()`

3.1 Dart 3 Programming — Sound Null Safety, Records, Sealed Classes, `Future`/`Stream` & `Isolate.run` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Dart 3 Programming — Sound Null Safety, Records, Sealed Classes, `Future`/`Stream` & `Isolate.run` in real Flutter systems — specifically: Master...

3.2 Flutter Widgets — `StatelessWidget` vs `StatefulWidget` (`initState`, `didUpdateWidget`, `dispose`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Widgets — `StatelessWidget` vs `StatefulWidget` (`initState`, `didUpdateWidget`, `dispose`) in real Flutter systems — specifically: Unders...

Module 3 Hands-On Capstone Project (100 Marks): Dart 3 Sealed Quiz Result Engine & Lifecycle-Safe Stateful Counter Model — Model asynchronous quiz evaluation states (`Loading`, `Completed`, `Failure`) using a Dart 3 `sealed class` with Record destructuring, and pair it with a lifecycle guard preventing `setState...

MODULE 4 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 4: Flutter's Golden Layout Rule (`BoxConstraints`) & Material `Scaffold` Anatomy

Internalize Flutter's single-pass layout protocol ("Constraints go down. Sizes go up. Parent sets position.") with `Container`, `Padding`, `SizedBox`, `LayoutBuilder`, and `Scaffold` (`AppBar`, `SafeArea`, `SnackBar`).

Objectives: ✓ Predict tight vs loose vs unbounded `BoxConstraints` across `Center`, `Align`, `SizedBox`, and `ConstrainedBox` • ✓ Build responsive mobile/tablet breakpoints cleanly with `LayoutBuilder` rather than global `MediaQuery` rebuilds • ✓ Structure Material 3 screens with `Scaffold`, `AppBar`,

`FloatingActionButton`, `NavigationBar`, `SafeArea`, and `ScaffoldMessenger`

4.1 Flutter Layouts — `BoxConstraints` ("Constraints Down, Sizes Up"), `SizedBox` & `LayoutBuilder` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Layouts — `BoxConstraints` ("Constraints Down, Sizes Up"), `SizedBox` & `LayoutBuilder` in real Flutter systems — specifically: How Flutte...

4.2 Flutter Scaffold — `AppBar`, `SafeArea`, `NavigationBar`, `resizeToAvoidBottomInset` & `ScaffoldMessenger` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Scaffold — `AppBar`, `SafeArea`, `NavigationBar`, `resizeToAvoidBottomInset` & `ScaffoldMessenger` in real Flutter systems — specifically:...

Module 4 Hands-On Capstone Project (100 Marks): BoxConstraints Solver & Adaptive Mobile/Tablet Scaffold Shell — Build a `BoxConstraints` clamping solver (`minWidth`, `maxWidth`, `minHeight`, `maxHeight`) paired with a responsive `LayoutBuilder` breakpoint selector switching between a mobile bottom `Na...

MODULE 5 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 5: Flex Layouts (`Row`, `Column`, `Expanded`, `Flexible`) & Layered `Stack` Positioning

Eliminate Flutter's notorious yellow-and-black `RenderFlex overflowed` stripes and `Vertical viewport was given unbounded height` errors using `Expanded`, `Flexible`, `Wrap`, and `Stack` + `Positioned`.

Objectives: ✓ Distinguish `MainAxisAlignment` vs `CrossAxisAlignment` across horizontal `Row` and vertical `Column` flex containers • ✓ Solve `RenderFlex overflowed` and unbounded `ListView`-in-`Column` crashes using `Expanded` (`FlexFit.tight`) vs `Flexible` (`FlexFit.loose`) • ✓ Layer badges, video overlays, and avatars cleanly using `Stack`, `Positioned`, `Align`, and `IndexedStack`

5.1 Row & Column (`Flex`) — `MainAxis` vs `CrossAxis`, `Expanded` vs `Flexible` & Overflow Prevention [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Row & Column (`Flex`) — `MainAxis` vs `CrossAxis`, `Expanded` vs `Flexible` & Overflow Prevention in real Flutter systems — specifically: Master F...

5.2 Flutter Stack & `Positioned` — Non-Positioned Sizing, `Clip.none`, Hit-Testing & `IndexedStack` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Stack & `Positioned` — Non-Positioned Sizing, `Clip.none`, Hit-Testing & `IndexedStack` in real Flutter systems — specifically: Overlap wi...

Module 5 Hands-On Capstone Project (100 Marks): Flex Space Allocator & `IndexedStack` State-Preserving Tab Switcher — Build a two-phase `RenderFlex` layout simulator computing exact pixel widths for fixed and `Expanded(flex: k)` children while flagging overflow pixels when fixed children exceed `maxWidth`,...

MODULE 6 OF 16 • 12 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 6: Interactive Text Input (`TextField`, `FocusNode`) & Material 3 Buttons

Capture user text cleanly with `TextField`, `TextEditingController`, `FocusNode`, and `InputFormatter`s, and apply Material 3's button hierarchy (`FilledButton`, `OutlinedButton`, `TextButton`, `IconButton`).

Objectives: ✓ Create and dispose `TextEditingController` and `FocusNode` instances inside `State` (never inside `build()`) • ✓ Configure `TextInputType`, `TextInputAction.next`, `obscureText`, `InputDecoration`, and `FilteringTextInputFormatter` • ✓ Implement Material 3 button emphasis hierarchy (`FilledButton` -> `ElevatedButton` -> `OutlinedButton` -> `TextButton`) and async double-tap locks

6.1 Flutter TextField — `TextEditingController`, `FocusNode`, `InputDecoration` & Memory Disposal [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter TextField — `TextEditingController`, `FocusNode`, `InputDecoration` & Memory Disposal in real Flutter systems — specifically: Control text...

6.2 Flutter Material 3 Buttons — `FilledButton`, `OutlinedButton`, `WidgetStateProperty` & Loading Locks [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Material 3 Buttons — `FilledButton`, `OutlinedButton`, `WidgetStateProperty` & Loading Locks in real Flutter systems — specifically: Choos...

Module 6 Hands-On Capstone Project (100 Marks): Debounced Search Input Controller & Idempotent Submit Button Guard — Engineer a leak-safe search input state controller that sanitizes input, ignores duplicate query emissions, and pairs with an async submit button state machine that disables `onPressed` (`nu...

MODULE 7 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 7: Declarative Form Validation (`GlobalKey<FormState>`) & Lazy Virtualized Lists

Validate multi-field forms cleanly using `Form`, `GlobalKey<FormState>`, and `TextFormField` validators, and render 50,000+ rows smoothly using lazy `ListView.builder` / `SliverList` virtualization.

Objectives: ✓ Coordinate multi-field validation and `AutovalidateMode.onUserInteraction` via `GlobalKey<FormState>` • ✓ Contrast eager `ListView(children: [...])` with lazy viewport-virtualized `ListView.builder` and `ListView.separated` • ✓ Supply `itemExtent` / `prototypeItem` so Flutter computes scrollbar metrics in O(1) instead of measuring offscreen items

7.1 Flutter Forms — `Form`, `GlobalKey<FormState>`, `TextFormField` Validators & `AutovalidateMode` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Forms — `Form`, `GlobalKey<FormState>`, `TextFormField` Validators & `AutovalidateMode` in real Flutter systems — specifically: Group mult...

7.2 Flutter Lists — `ListView.builder`, `ListView.separated`, `itemExtent`, `ValueKey` & Slivers [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Lists — `ListView.builder`, `ListView.separated`, `itemExtent`, `ValueKey` & Slivers in real Flutter systems — specifically: Render massiv...

Module 7 Hands-On Capstone Project (100 Marks): Multi-Field Enrollment Validator & Virtualized Sliver Viewport Window Calculator — Build a composable `FormFieldValidator<String>` pipeline (email, password strength, confirm match) and a virtualized `ListView.builder` viewport calculator computing exact visible start/end...

MODULE 8 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 8: Responsive `GridView` Delegates & Coordinated `TabBar` + `TabBarView` Navigation

Build responsive 2D product/course grids with `SliverGridDelegateWithMaxCrossAxisExtent` and synchronized swipeable tabs using `DefaultTabController`, `TabController`, and `AutomaticKeepAliveClientMixin`.

Objectives: ✓ Choose `SliverGridDelegateWithMaxCrossAxisExtent` over hardcoded column counts so grids scale from phones to foldables/desktop • ✓ Control grid cell proportions cleanly via `childAspectRatio`, `mainAxisSpacing`, `crossAxisSpacing`, and `mainAxisExtent` • ✓ Synchronize `TabBar` and `TabBarView` with `DefaultTabController` or a custom `TickerProvider` `TabController` and preserve tab scroll state

8.1 Flutter GridView — `GridView.builder`, `MaxCrossAxisExtent` vs `FixedCrossAxisCount` & `mainAxisExtent` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter GridView — `GridView.builder`, `MaxCrossAxisExtent` vs `FixedCrossAxisCount` & `mainAxisExtent` in real Flutter systems — specifically: Co...

8.2 Flutter TabBar & `TabBarView` — `DefaultTabController`, `SingleTickerProviderStateMixin` & KeepAlive [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter TabBar & `TabBarView` — `DefaultTabController`, `SingleTickerProviderStateMixin` & KeepAlive in real Flutter systems — specifically: Wire...

Module 8 Hands-On Capstone Project (100 Marks): Responsive `MaxCrossAxisExtent` Grid Planner & `KeepAlive` Tab State Coordinator — Create a responsive grid geometry planner replicating Flutter's `SliverGridDelegateWithMaxCrossAxisExtent` column-count math across mobile, tablet, and desktop widths, paired with a `TabCont...

MODULE 9 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 9: Material 3 `NavigationDrawer` Routing & The Gesture Arena (`GestureDetector`, `Dismissible`)

Implement modal and permanent `NavigationDrawer` navigation shells cleanly, and master Flutter's pointer disambiguation **Gesture Arena** across `InkWell`, `GestureDetector`, `Dismissible`, and `InteractiveViewer`.

Objectives: ✓ Build Material 3 `NavigationDrawer` menus with `selectedIndex` highlighting and proper drawer-close-before-push routing • ✓ Explain how competing `GestureRecognizer`s resolve tap vs drag vs scale conflicts inside Flutter's Gesture Arena • ✓ Implement `Dismissible` swipe-to-delete rows safely using unique `ValueKey`s (never list array index keys!)

9.1 Flutter NavigationDrawer — Material 3 `NavigationDrawerDestination`, Drawer Closing & Routing [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter NavigationDrawer — Material 3 `NavigationDrawerDestination`, Drawer Closing & Routing in real Flutter systems — specifically: Upgrade lega...

9.2 Flutter Gestures — The Gesture Arena, `GestureDetector` (`HitTestBehavior`), `InkWell` & `Dismissible` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter Gestures — The Gesture Arena, `GestureDetector` (`HitTestBehavior`), `InkWell` & `Dismissible` in real Flutter systems — specifically: Und...

Module 9 Hands-On Capstone Project (100 Marks): Safe `Dismissible` Key Manager & Gesture Arena Disambiguator — Build a gesture disambiguation model resolving competing tap vs horizontal-drag pointer sequences based on touch-slop thresholds (`kTouchSlop = 18.0`), paired with a `ValueKey`-verified `Dis...

MODULE 10 OF 16 • 11 LIVE + LAB HOURS • 2 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 10: Scalable State Management (`ValueNotifier`, Riverpod, BLoC) & Flutter vs React Native

Architect production apps with unidirectional data flow (`UI = f(state)`) using `ValueNotifier`, `ChangeNotifier`, **Riverpod** (`NotifierProvider`, `AsyncNotifier`), and **BLoC**, and evaluate Flutter vs React Native trade-offs objectively.

Objectives: ✓ Separate ephemeral local UI state (`StatefulWidget`) from shared app/domain state (`Riverpod` / `BLoC` / `Provider`) • ✓ Eliminate unnecessary whole-screen rebuilds using granular `ref.watch(provider.select((s) => s.field))` selectors • ✓ Compare Flutter (Dart AOT + Impeller custom canvas) vs React Native (TypeScript/Hermes + JSI/Fabric native OEM views) in principal architecture interviews

10.1 Flutter State Management — `ValueNotifier`, `InheritedWidget`, Provider, Riverpod & BLoC [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter State Management — `ValueNotifier`, `InheritedWidget`, Provider, Riverpod & BLoC in real Flutter systems — specifically: Progress from bui...

10.2 Flutter vs React Native — Impeller Canvas vs JSI/Fabric Native Views, OTA & Architecture Trade-Offs [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flutter vs React Native — Impeller Canvas vs JSI/Fabric Native Views, OTA & Architecture Trade-Offs in real Flutter systems — specifically: Object...

Module 10 Hands-On Capstone Project (100 Marks): Immutable Unidirectional Cart & Exam Progress Store with Granular Selectors — Implement a compile-safe immutable state store (`Notifier<ExamProgressState>`) with `copyWith` transitions and selector-filtered listener notifications that fire ONLY when the selected slice...

MODULE 11 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Scalable Flutter Architecture: Riverpod/BLoC, Freezed & GoRouter

Production-grade Flutter Training engineering module covering Scalable Flutter Architecture: Riverpod/BLoC, Freezed & GoRouter with hands-on labs and capstone build.

Objectives: ✓ Feature-First Clean Architecture & Immutable Dart 3 Sealed/Record Models — Purpose: Model exhaustive AsyncValue/Result states so null crashes never reach users. • ✓ Riverpod AsyncNotifier / BLoC Concurrency Transformers (droppable/restartable) — Purpose: Cancel stale search queries and prevent double-tap duplicate form submissions. • ✓ Declarative GoRouter Deep Links, ShellRoutes & Auth Redirect Guards — Purpose: Support universal iOS/Android web links while preserving bottom-tab navigation stacks. • ✓ Dio Interceptors, Queued JWT Refresh & Typed Retrofit/JsonSerializable — Purpose: Centralize network retries, error mapping, and zero-reflection JSON parsing.

11.1 Feature-First Clean Architecture & Immutable Dart 3 Sealed/Record Models [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Feature-First Clean Architecture & Immutable Dart 3 Sealed/Record Models in real Flutter systems — specifically: Model exhaustive AsyncValue/Resul...

11.2 Riverpod AsyncNotifier / BLoC Concurrency Transformers (droppable/restartable) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Riverpod AsyncNotifier / BLoC Concurrency Transformers (droppable/restartable) in real Flutter systems — specifically: Cancel stale search queries...

11.3 Declarative GoRouter Deep Links, ShellRoutes & Auth Redirect Guards [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Declarative GoRouter Deep Links, ShellRoutes & Auth Redirect Guards in real Flutter systems — specifically: Support universal iOS/Android web link...

11.4 Dio Interceptors, Queued JWT Refresh & Typed Retrofit/JsonSerializable [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Dio Interceptors, Queued JWT Refresh & Typed Retrofit/JsonSerializable in real Flutter systems — specifically: Centralize network retries, error m...

Module 11 Hands-On Capstone Project (100 Marks): Modular Cross-Platform SaaS Super-App Shell in Flutter — Design, implement, test, and document 'Modular Cross-Platform SaaS Super-App Shell in Flutter' applying every production concept from Module 11 (Scalable Flutter Architecture: Riverpod/BLoC,...

MODULE 12 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Offline-First Local Databases (Drift/SQLite), Secure Storage & Sync

Production-grade Flutter Training engineering module covering Offline-First Local Databases (Drift/SQLite), Secure Storage & Sync with hands-on labs and capstone build.

Objectives: ✓ Reactive Local SQL with Drift (SQLite), Indexes & Schema Migrations — Purpose: Stream live local queries to widgets and migrate mobile DB tables safely across app updates. • ✓ Optimistic Local Writes + Background Outbox Conflict Resolution — Purpose: Let field users work 100% offline and sync queued mutations cleanly upon reconnect. • ✓ flutter_secure_storage (iOS Keychain / Android Keystore) & Biometrics — Purpose: Store refresh tokens in hardware-encrypted enclaves unlocked via FaceID/Fingerprint. • ✓ Background Isolates (Isolate.run) for Heavy JSON/Image Processing — Purpose: Keep UI animations locked at 60/120 FPS while parsing large payloads off the main thread.

12.1 Reactive Local SQL with Drift (SQLite), Indexes & Schema Migrations [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Reactive Local SQL with Drift (SQLite), Indexes & Schema Migrations in real Flutter systems — specifically: Stream live local queries to widgets a...

12.2 Optimistic Local Writes + Background Outbox Conflict Resolution [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Optimistic Local Writes + Background Outbox Conflict Resolution in real Flutter systems — specifically: Let field users work 100% offline and sync...

12.3 flutter_secure_storage (iOS Keychain / Android Keystore) & Biometrics [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply flutter_secure_storage (iOS Keychain / Android Keystore) & Biometrics in real Flutter systems — specifically: Store refresh tokens in hardware-enc...

12.4 Background Isolates (Isolate.run) for Heavy JSON/Image Processing [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Background Isolates (Isolate.run) for Heavy JSON/Image Processing in real Flutter systems — specifically: Keep UI animations locked at 60/120 FPS...

Module 12 Hands-On Capstone Project (100 Marks): Offline-Capable Delivery & Inspection App for iOS + Android — Design, implement, test, and document 'Offline-Capable Delivery & Inspection App for iOS + Android' applying every production concept from Module 12 (Offline-First Local Databases (Drift/SQL...

MODULE 13 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Platform Channels (MethodChannel), Impeller Rendering & Sliver Tuning

Production-grade Flutter Training engineering module covering Platform Channels (MethodChannel), Impeller Rendering & Sliver Tuning with hands-on labs and capstone build.

Objectives: ✓ Type-Safe Pigeon / MethodChannel & EventChannel Native Bridges — Purpose: Invoke native Kotlin/Swift payment, scanner, or Bluetooth APIs from Dart cleanly. • ✓ CustomScrollView Slivers, RepaintBoundary & const Widget Discipline — Purpose: Eliminate raster/UI thread jank on media-rich e-commerce feeds. • ✓ Cached Image Sizing (memCacheWidth), Shimmer Skeletons & Memory Audits — Purpose: Prevent iOS/Android out-of-memory kills when scrolling high-res photo galleries. • ✓ Push Notifications, Local Schedules, Camera/Files & Adaptive Tablet UI — Purpose: Deliver responsive layouts that shine on compact phones, foldables, and iPads alike.

13.1 Type-Safe Pigeon / MethodChannel & EventChannel Native Bridges [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Type-Safe Pigeon / MethodChannel & EventChannel Native Bridges in real Flutter systems — specifically: Invoke native Kotlin/Swift payment, scanner...

13.2 CustomScrollView Slivers, RepaintBoundary & const Widget Discipline [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply CustomScrollView Slivers, RepaintBoundary & const Widget Discipline in real Flutter systems — specifically: Eliminate raster/UI thread jank on med...

13.3 Cached Image Sizing (memCacheWidth), Shimmer Skeletons & Memory Audits [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Cached Image Sizing (memCacheWidth), Shimmer Skeletons & Memory Audits in real Flutter systems — specifically: Prevent iOS/Android out-of-memory k...

13.4 Push Notifications, Local Schedules, Camera/Files & Adaptive Tablet UI [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Push Notifications, Local Schedules, Camera/Files & Adaptive Tablet UI in real Flutter systems — specifically: Deliver responsive layouts that shi...

Module 13 Hands-On Capstone Project (100 Marks): Smooth 120fps Custom Sliver Storefront with Native SDK Bridge — Design, implement, test, and document 'Smooth 120fps Custom Sliver Storefront with Native SDK Bridge' applying every production concept from Module 13 (Platform Channels (MethodChannel), Imp...

MODULE 14 OF 16 • 16 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Widget/Golden Tests, Integration Suites, Flavors & App/Play Store CI

Production-grade Flutter Training engineering module covering Widget/Golden Tests, Integration Suites, Flavors & App/Play Store CI with hands-on labs and capstone build.

Objectives: ✓ Unit, Widget & Pixel-Accurate Golden Screenshot Regression Tests — Purpose: Catch broken layouts and state bugs across screen sizes in fast headless CI runs. • ✓ End-to-End Device Integration Testing (integration_test / Patrol) — Purpose: Automate real tap-and-scroll checkout journeys on emulators before store submission. • ✓ Build Flavors (--dart-define-from-file), Obfuscation & Split Debug Info — Purpose: Maintain separate dev/staging/prod icons, API URLs, and obfuscated Dart binaries. • ✓ Codemagic / GitHub Actions + Fastlane Signing for TestFlight & Play Store — Purpose: Automate code signing, Crashlytics symbol upload, and dual-store publishing.

14.1 Unit, Widget & Pixel-Accurate Golden Screenshot Regression Tests [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Unit, Widget & Pixel-Accurate Golden Screenshot Regression Tests in real Flutter systems — specifically: Catch broken layouts and state bugs acros...

14.2 End-to-End Device Integration Testing (integration_test / Patrol) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply End-to-End Device Integration Testing (integration_test / Patrol) in real Flutter systems — specifically: Automate real tap-and-scroll checkout jo...

14.3 Build Flavors (--dart-define-from-file), Obfuscation & Split Debug Info [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Build Flavors (--dart-define-from-file), Obfuscation & Split Debug Info in real Flutter systems — specifically: Maintain separate dev/staging/prod...

14.4 Codemagic / GitHub Actions + Fastlane Signing for TestFlight & Play Store [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Codemagic / GitHub Actions + Fastlane Signing for TestFlight & Play Store in real Flutter systems — specifically: Automate code signing, Crashlyti...

Module 14 Hands-On Capstone Project (100 Marks): Multi-Flavor (Dev/Stage/Prod) Automated iOS & Android Release — Design, implement, test, and document 'Multi-Flavor (Dev/Stage/Prod) Automated iOS & Android Release' applying every production concept from Module 14 (Widget/Golden Tests, Integration Suite...

MODULE 15 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Offline-First Flutter Architecture, Isolates & Impeller 60/120fps Tuning

Self-explanatory list-and-bullet production mastery module covering Offline-First Flutter Architecture, Isolates & Impeller 60/120fps Tuning in Flutter Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Clean Layered Architecture (Presentation, Domain, Data) with AsyncNotifier & Sealed States — Purpose: Separate UI widgets cleanly from API/SQLite repositories for testable, scalable Flutter codebases. • ✓ Dart Isolates (Isolate.run) for Heavy JSON/Crypto Work & Sliver Viewport Optimization — Purpose: Keep UI animations locked at 60/120fps under the Impeller engine with zero main-isolate jank. • ✓ Secure Token Storage, Dio Refresh-Token Queues & GoRouter Deep-Link Guards — Purpose: Handle expired sessions seamlessly and route authenticated deep links across Android, iOS, and Web. • ✓ Build, verify, and document 'Smooth 120fps Offline-Capable Commerce App with Riverpod & Background Isolates' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

15.1 Clean Layered Architecture (Presentation, Domain, Data) with AsyncNotifier & Sealed States [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Clean Layered Architecture (Presentation, Domain, Data) with AsyncNotifier & Sealed States in real Flutter systems — specifically: Separate UI wid...

15.2 Dart Isolates (Isolate.run) for Heavy JSON/Crypto Work & Sliver Viewport Optimization [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Dart Isolates (Isolate.run) for Heavy JSON/Crypto Work & Sliver Viewport Optimization in real Flutter systems — specifically: Keep UI animations l...

15.3 Secure Token Storage, Dio Refresh-Token Queues & GoRouter Deep-Link Guards [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Secure Token Storage, Dio Refresh-Token Queues & GoRouter Deep-Link Guards in real Flutter systems — specifically: Handle expired sessions seamles...

Module 15 Hands-On Capstone Project (100 Marks): Smooth 120fps Offline-Capable Commerce App with Riverpod & Background Isolates — Architect, implement, test, and document 'Smooth 120fps Offline-Capable Commerce App with Riverpod & Background Isolates' applying every bullet-wise engineering concept from Module 15 (Offli...

MODULE 16 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Flutter Widget/Golden Testing, DevTools Profiling & Store Release CI/CD

Self-explanatory list-and-bullet production mastery module covering Flutter Widget/Golden Testing, DevTools Profiling & Store Release CI/CD in Flutter Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken MockAttempt AI Viva.

Objectives: ✓ Unit, Widget & Golden Visual Regression Testing with Mocktail in Flutter — Purpose: Catch layout overflows, broken dark/light themes, and state regressions before every release. • ✓ Flutter DevTools Memory/CPU Profiling, Widget Rebuild Audits & Crashlytics — Purpose: Eliminate stream/controller leaks, excessive rebuilds, and production crashes proactively. • ✓ Multi-Environment Flavors, Code Signing, Shorebird/OTA Concepts & Flutter Interviews — Purpose: Explain Widget-Element-RenderObject trees, keys, and state trade-offs in technical interviews. • ✓ Build, verify, and document 'Automated Widget + Integration Test Suite & Multi-Flavor Release Pipeline' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + MockAttempt AI Technical Interviews.

16.1 Unit, Widget & Golden Visual Regression Testing with Mocktail in Flutter [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Unit, Widget & Golden Visual Regression Testing with Mocktail in Flutter in real Flutter systems — specifically: Catch layout overflows, broken da...

16.2 Flutter DevTools Memory/CPU Profiling, Widget Rebuild Audits & Crashlytics [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Flutter DevTools Memory/CPU Profiling, Widget Rebuild Audits & Crashlytics in real Flutter systems — specifically: Eliminate stream/controller lea...

16.3 Multi-Environment Flavors, Code Signing, Shorebird/OTA Concepts & Flutter Interviews [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Multi-Environment Flavors, Code Signing, Shorebird/OTA Concepts & Flutter Interviews in real Flutter systems — specifically: Explain Widget-Elemen...

Module 16 Hands-On Capstone Project (100 Marks): Automated Widget + Integration Test Suite & Multi-Flavor Release Pipeline — Architect, implement, test, and document 'Automated Widget + Integration Test Suite & Multi-Flavor Release Pipeline' applying every bullet-wise engineering concept from Module 16 (Flutter Wi...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: QuickMed Cross-Platform Telehealth Booking & Prescription Tracker App • **Confirmed Role:** Flutter & Dart Cross-Platform Mobile Intern

Domain & Production Infra: Digital Healthcare & Diagnostics Consumer App Client (Flutter 3 + Dart 3 + Riverpod/BLoC + Dio REST + Local Secure Storage (Android & iOS/Web))

- Crafted 60fps adaptive Flutter UI flows with custom slivers, animations, and responsive design tokens.
- Architected predictable state management with Riverpod/BLoC, clean repository layers, and Dart 3 sealed classes.
- Connected Dio HTTP interceptors, secure JWT keychain storage, push notifications, and offline caching.
- Published release-ready cross-platform builds validated with widget and integration test suites.

Aligned Global Vendor Certifications: **AFD-200** (Flutter Certified Application Developer (AFD-200)) • **GOOGLE-AAD** (Android Kotlin & Jetpack Compose Developer Credential)

Target Job Roles: Flutter Mobile Developer (iOS & Android), Cross-Platform Mobile Engineer, Dart & Flutter Product Developer, Mobile Application SDE-1

Realistic Annual CTC Scope: **Fresher / Entry-Level (0–1 Yr):** ₹4.8 LPA – ₹9.2 LPA **Mid-Level Flutter Product Engineer (1–3 Yrs):** ₹9.5 LPA – ₹19.5 LPA **Senior Cross-Platform Mobile Architect (3–6+ Yrs):** ₹20.0 LPA – ₹38.0+ LPA