

Essential Understanding of Mobile App Development -

A 3-Day Course for Non-CS Professionals

DAY 1: Mobile App Basics & How They Work

What is a Mobile App?

- - Native, Hybrid, and Web apps – with examples (e.g., WhatsApp, Amazon)
- - Platforms: Android, iOS, Flutter – what they are and differences
- - App installation, updates, and uninstallation – what happens technically

Mobile App Architecture

- - Frontend (UI), Backend (Logic), and Database (Storage) explained
- - How frontend talks to backend using APIs

Data Flow & How Apps Communicate

- - Clicking Login: request-response cycle
- - API calls, HTTP/HTTPS, GET/POST methods – simplified
- - JSON: how data travels

Tools & Languages Commonly Used

- - Flutter, React Native, Kotlin, Swift – roles
- - Backend tech: Node.js, Firebase, PHP – purpose only
- - Third-party services: Google Maps, Razorpay, Twilio

DAY 2: Core App Features and Technical Concepts

Authentication & User Management

- - Login, OTP, Google/Facebook Sign-In – flow overview
- - Session, token, cookies – in layman terms
- - Secure authentication – what can go wrong

Common App Features

- - Push notifications – how they work
- - Location, camera, gallery, microphone access
- - Background tasks – e.g., downloading in background

User Interface (UI) & User Experience (UX)

- - Good vs. Bad UI examples
- - Design tools (Figma, Adobe XD), navigation patterns

App Permissions & Security Basics

- - Why apps ask for permissions
- - Data exposure risks and best practices
- - SSL, HTTPS, encryption explained simply

DAY 3: Development Process, Testing, Deployment & Management

App Development Lifecycle

- - Requirement → Design → Development → Testing → Deployment
- - Roles: UI/UX Designer, Developer, QA, Product Manager
- - Agile, Sprint, Scrum in simple terms

APIs & 3rd Party Integrations

- - What is API integration? Examples: Razorpay, Google Maps, Firebase
- - Public vs Private APIs
- - API documentation tools (Postman, Swagger)

Testing & Bug Fixing

User Acceptance Testing (UAT) for Mobile Apps

- What is UAT and why it matters: Ensuring the app works for the end user
- Who performs UAT: Involvement of business users, clients, or product owners
- UAT vs Developer Testing vs QA Testing – explained simply
- How to define Acceptance Criteria using user stories (INVEST model)
- Example: "As a user, I want to log in using OTP so that I can access my account securely"
- Designing UAT test cases from real-life scenarios (login, payment, notifications)
- When to do UAT: Before production release or app store submission
- Tools commonly used: Spreadsheets, JIRA, Zephyr (just awareness level)
- Common UAT issues: UI mismatches, wrong data shown, flows not working as expected
- Tips to report UAT bugs clearly: screenshot + steps + expected vs actual
- Sign-off process: What to verify before accepting delivery from the dev team

Deployment and Maintenance

- - Publishing apps to Play Store & App Store
- - Versioning, updates, rollouts
- - Crash logs, analytics, user feedback

Communicating with Developers

- - Common terms explained: 'Build broken', 'Merge conflict', etc.
- - How to give clear requirements
- - Red flags to look for in project updates