Mobile App Performance Optimization

Mobile Dev Guide · Module 7 of 8 · CHERIEDU Dev Series

1. Why Performance Is a Feature

53% of users abandon apps that take more than 3 seconds to load. Performance is not a nice-to-have — it directly affects user retention, store ratings, and revenue.

2. Flutter Performance Golden Rules

1. Use const constructors wherever possible — Flutter skips rebuilding const widgets.
2. Never do heavy work in build() — use initState() or async methods.
3. Use ListView.builder (lazy loading) not ListView with all items at once.
4. Cache network images with cached_network_image package.
5. Use RepaintBoundary to isolate expensive animations.

3. Memory Management

- Dispose controllers in dispose(): TextEditingController, AnimationController.
- Cancel stream subscriptions in dispose() to prevent memory leaks.
- Use weak references for listeners where possible.
- Profile memory with Flutter DevTools — look for increasing memory over time.

4. Network Performance

Strategy	Benefit	Implementation
Pagination	Load 20 items at a time	API: ?page=1&limit=20
Image compression	Faster loading	WebP format, thumbnail URLs
Response caching	Fewer API calls	Cache with expiry time
Optimistic UI	Feels instant	Update UI before API confirms

5. Profiling With Flutter DevTools

- Run: flutter run --profile — enables performance profiling mode.
- Open DevTools: flutter pub global run devtools in terminal.
- Performance tab: Shows frame rendering time. Aim for 16ms per frame (60 FPS).
- Memory tab: Track heap usage and identify leaks.
- CPU Profiler: Find which methods take the most time.

OPTIMISE

Profile your CHERI SMS Flutter app. Identify the 3 slowest screens. Apply at least 2 optimisations to each. Measure before and after — document the improvement in milliseconds.