



Memory Performance Analysis (Flutter DevTools)

Test Environment

- Device: Android physical device
- Build Mode: Debug mode
- Tool Used: Flutter DevTools → Memory Tab
- Dataset Size: 50,000+ songs
- Test Duration: Continuous scrolling for several minutes

Test Procedure

1. Application launch kiya.
2. Music library screen open ki.
3. Infinite scroll repeatedly perform kiya.
4. Multiple search operations perform kiye.
5. Different tabs switch kiye (Home, Songs, Playlist, Details).

6. Audio playback run kiya background me.
 7. Memory usage continuously DevTools me monitor kiya.
-

Observed Memory Behavior

- Initial memory usage: ~350 MB
 - Peak memory usage: ~420 MB
 - Continuous scrolling ke baad memory stable ho gayi.
 - No continuous linear growth observed.
 - Garbage collection events regularly visible.
 - Memory spikes temporary the (data fetch time).
 - After GC memory wapas baseline ke paas aa gayi.
-

Stability Result

Application ne large dataset ke saath stable memory maintain ki.

- ✓ No memory leak detected
 - ✓ No unbounded growth
 - ✓ UI smooth raha during heavy scroll
 - ✓ Repeated search se memory increase permanent nahi hua
 - ✓ Lazy loading properly working
-

Why Memory Stable Hai

App memory efficient hai because:

- ListView.builder lazy widgets create karta hai
- Visible items hi memory me hote hain
- compute() isolate JSON parsing main thread block nahi karta
- Pagination se limited data fetch hota hai

- Duplicate tracks prevent karne ke liye Set use hota hai
 - Garbage collection automatically unused memory free karta hai
-

Technical Conclusion

The application successfully handles large-scale data rendering (50,000+ items) while maintaining stable heap memory usage. The memory profile confirms efficient lazy loading, controlled pagination, and proper garbage collection behavior.

Final Result

- Memory usage stable under heavy load
- Suitable for large dataset virtualization
- No performance degradation observed