

# Bugs & Correctness Rubric

CRITERION	1	2	3	4	5
Error handling	No try/catch, unhandled promise rejections crash the app	Some try/catch but errors silently swallowed or logged inconsistently	Errors caught at major boundaries, user sees generic fallback	Error boundaries per feature, structured logging, user-friendly messages	Typed error hierarchy, automatic retry/recovery, error tracking in CI
State management	Race conditions in fetches, stale closures, prop-drilling 5+ levels	Some shared state but updates conflict, no derived state patterns	Single store or context per domain, basic optimistic updates	Derived state computed, no stale reads, concurrent-safe updates	State machines for complex flows, snapshot testing of transitions
Edge cases	Null/undefined crashes, empty arrays render broken UI, no loading states	Some null checks but inconsistent, happy path only in tests	Empty states handled, boundary values tested, basic guards	Exhaustive switch/case, property-based tests for edge inputs	Fuzzing or generative tests, zero known uncovered branches
Type safety	Widespread `any`, no strict mode, runtime type mismatches	Some types but frequent `as` casts and non-null assertions	Strict mode on, most interfaces typed, few escape hatches	Discriminated unions, exhaustive narrowing, no unsafe casts	End-to-end type safety (API to UI), generics for reusable code

Code Consultant