Laravel: Cleaner Code Practices ✨

Clean code isn't a luxury—it's how we move fast without breaking things. In Laravel, a few simple habits can keep your app easy to read, test, and scale.

1) Keep controllers slim by delegating to services □
Controllers should coordinate, not calculate.
✓□ Handle request/response, validation, and route model binding.
✓□ Put business rules in service classes (e.g., BillingService).
✓□ Benefit: smaller files, easier unit tests, clearer ownership.

2) Use middleware only where it's needed ⚖□
Middleware is powerful, but overusing it creates noise.
✓□ Apply per-route or per-group instead of globally when possible.
✓□ Reserve it for cross-cutting concerns like auth, rate limiting, or logging.
✓□ Tip: name middleware clearly so intent is obvious (ensure.role:admin).

3) Stay consistent with naming 🛤
Predictable names reduce cognitive load.
✓□ Models: singular (User, Order); tables: plural (users, orders).
✓□ Pivots: alphabetical (order_user, not user_order).
✓□ Relationships: describe what they return (user->orders(), not getData()).

4) Refactor regularly, don't just patch □
Tiny quick fixes become big messes later.
✓□ Schedule "cleanup sprints" to remove dead code and duplicated logic.
✓□ Split long methods; extract small functions with clear names.
✓□ Add lightweight tests around tricky parts before changing them.
✓□ Result: fewer surprises, fewer regressions.

5) Document key architectural decisions 📝
You don't need a novel—just clarity.
✓□ Record why you chose an approach (queues vs. sync jobs, repo vs. direct model).
✓□ Keep short ADRs (Architecture Decision Records) in your repo.
✓□ Update the README with module boundaries and naming rules.
✓□ New teammates onboard faster; old teammates forget less.

Bottom line ✅
Cleaner code is a daily habit: small, consistent choices that make the next change safer and faster. Your future self (and your team) will thank you.