Senior Developer Starter Kit

A fast, focused way to level up from Middle to Senior Developer.

What Makes a True Senior Developer?

Many developers think seniority is about "years in the role." But real seniority is shown through:

- **Problem-solving** under pressure
- **Clear communication** across teams
- **Architectural thinking** for scalability
- **Ownership** beyond the code
- **Relentless learning** and self-direction

> 💬 *“I used to think being a senior meant writing perfect code. Now I know it means making the whole system — and team — better.”*

This kit gives you a clear path forward.

✅ Senior Developer Skill Checklist

👨‍💻 Technical Skills

- Solve unfamiliar problems without hand-holding
- Write scalable, maintainable, **boring** code
- Weigh trade-offs and explain your reasoning
- Refactor safely, with confidence
- Write and trust unit, integration, and e2e tests

🗣️ Communication Skills

- Explain technical ideas to non-devs (e.g. PMs)
- Give and receive feedback in code reviews
- Justify your architecture respectfully
- Break big features into small, clear tasks

> 💬 *"Can you explain your solution in 2 minutes to a junior, and in 10 seconds to a product manager? That's senior communication."*

🌱 Leadership & Ownership

- Mentor junior devs without micromanaging
- Proactively fix broken processes or tech debt
- Raise concerns before they become blockers
- Push for best practices, even under pressure

🎯 Senior Interview Topics: What You Should Be Ready to Discuss

- How you’ve improved team velocity (real example)
- A mistake you made and how you handled it
- How you help unblock others
- When you chose one architecture over another — and why
- How you prioritize under deadlines
- Trade-offs between monolith vs microservices
- Testing strategy: when to test, what to skip
- Algorithm basics: arrays, recursion, time complexity
- Patterns: Singleton, Observer, Factory, etc.

> 💬 *“Senior interviews aren’t about what you’ve memorized — they’re about how you reason through unclear problems.”*

🗣️ Communicate Like a Senior Engineer

Before a meeting:

> *"Here's what I plan to cover and why it matters to the team."*

When proposing an idea:

> *"I think this approach reduces complexity and makes onboarding easier."*

In code reviews:

> *"I wonder if we could simplify this logic — maybe by extracting a helper?"*

When receiving feedback:

> *"Thanks — that’s a good point. Let me rethink that part."*

Good communication is **30% what you know, 70% how you share it.**

📅 Your 30-Day “Senior Level Up” Plan

Build senior-level habits in just 1 hour a day:

Day	Task
1–7	Read 1 architecture case study per day (from blog posts or open source)
8–14	Refactor a small personal project with senior-level thinking
15–21	Write a summary of a past project: what worked, what didn’t
22–28	Practice explaining a recent bug or decision to a non-tech friend
29–30	Identify 2 weak areas and create a focused learning plan

🪞 Reflect: What's Your Next Step?

Spend the next 10 minutes writing down:

- What’s one thing I’ve been avoiding that would help me grow?
- Where do I struggle to explain myself clearly?
- What would it take to build something *on my own*, end-to-end?

> 💬 *“Most devs get stuck at the mid-level not because of lack of talent — but because no one shows them what to focus on.”*

🚀 Want Help Leveling Up?

If you’re serious about the next step, I’ve built a full **Middle → Senior Frontend Bootcamp** that covers:

- Interview prep
- Systems design
- Real-world architecture
- Communication
- Building a full solo SaaS from scratch

🧭 Start here:
Frontend Developer Middle-to-Senior Bootcamp »