

## Coding with a Sixth Sense: The Power of Intuition in Tech

By Dharshan S. Kotian

---

### Page 1: What Is Intuition — and Why It Matters in Tech

There have been countless times during my coding journey — whether while practicing DSA late at night before a coding contest, or stuck building a feature for a web project — where I had no clue what to do next. I'd be staring at the screen, trying every approach, and then suddenly... something just clicked. I couldn't explain it logically. I just *knew* what to try — and often, it worked.

That quiet, unexplainable “gut feeling”? That's **intuition** — the inner voice that guides you even when your conscious mind hasn't figured everything out.

In India, we often grow up being told to “study hard” and “understand everything before applying.” And while understanding is important, in tech — especially in fast-paced coding environments — I've realised that **intuition is a silent superpower**. It fills the gap between what you've fully understood and what you've deeply absorbed through experience.

We don't talk about it much in tech circles, but we should — because I truly believe **intuition has saved me more times than I can count**.

---

### Page 2: How Intuition Works in Programming

Let's break it down. Intuition in tech isn't some magic ability. It's more like **muscle memory for your brain**. It's your mind quietly building patterns in the background — even when you feel lost.

#### It Shows Up in DSA

In the beginning, DSA problems felt like riddles in another language. I'd struggle for hours. But over time — after solving dozens of problems and failing at many — I started noticing patterns.

Eventually, I could glance at a problem and just *feel*:

“This is probably a dynamic programming question...”

“This input screams prefix sum...”

“I need a two-pointer approach here...”

Before I even knew the reason, **my mind had learned to recognise the structure**. That's intuition — built silently through repetition.

#### It Appears in Real-World Projects Too

While working on college projects or personal websites, I often found myself confused — wondering which

tool to pick, how to approach a layout, or why my logic wasn't working. But after doing this a few times, I started developing a sense:

- "This looks like an issue with how I structured my conditions."
- "Users might get confused here — I should simplify the flow."
- "This layout just doesn't feel user-friendly — let me try another."

I wasn't always sure why, but I had seen enough examples, read enough documentation, and failed enough times to just *sense* what might be wrong.

### ♥ It's in the Little Things Too

Intuition shows up when you: - Choose between two ways of writing the same loop. - Decide whether a function name makes sense. - Guess which line of code is causing a bug before running it.

These small instincts — built through hours of hands-on practice — eventually make your workflow smoother and faster.

It's not something you actively learn. It's something that *emerges* when you keep showing up, learning, building, and reflecting.

---

## Page 3: How to Master Intuition — and Use It Wisely

Here's the good news: **Intuition can be trained.**

Yes — you don't need to be a "genius" to build this sixth sense. You just need to **keep showing up**. Every late-night debugging session, every wrong approach, every "I don't get this" moment — your brain is learning, quietly.

### 👉 How to Build It: 1. Practice with Curiosity, Not Pressure

Don't try to understand everything perfectly from day one. Explore. Play. Build. Learn by doing. That's how intuition takes root.

#### 1. Reflect Often

Ask yourself: *Why did I think that would work?*

The more you analyse your own instincts, the sharper they become.

#### 2. Don't Fear Failure

Failure trains your gut better than success ever can. Every wrong turn makes your intuition more accurate next time.

#### 3. Expose Yourself to Patterns

Read others' code. Watch videos. Notice how certain problems or designs repeat. Your brain starts collecting these building blocks subconsciously.

#### 4. Trust Yourself, Even When You Don't Know Why Yet

Your mind is smarter than you think. Give it space to connect the dots.

---

##### **Bonus: How Intuition Saves You in Interviews**

One of the biggest advantages of intuition is how it **shines in high-pressure, time-bound situations** — like coding interviews, contests, or last-minute debugging.

When you're given a tough problem and the clock is ticking, **you don't have time to go line-by-line through every concept**. That's when your gut kicks in:

- You feel a familiar structure in the question.
- You sense which approach *won't* work.
- You guess the edge cases before they're even mentioned.

Many times in interviews, I've been able to **crack a problem I've never seen before** — simply because my intuition recognised fragments of it from past practice.

And it wasn't luck.

It was **quiet, slow, behind-the-scenes learning finally revealing itself**.

That's the beauty of intuition — it shows up when logic is overwhelmed.

---

##### **A Final Word — Especially for Beginners**

If you're a fresher just starting out — overwhelmed by DSA, confused by all the frameworks, unsure what half the buzzwords mean — I want to tell you something important:

**You're already building intuition.**

Even if it doesn't feel like it right now.

Even when you're stuck.

Even when you feel like giving up.

Your mind is watching. Absorbing. Learning.

Just keep going. One step at a time.

Someday, you'll look back at a problem, smile, and say:

"I don't know *how*, but I just *knew* what to do."

That's your sixth sense.

That's your silent superpower.

That's your *coding intuition*.

Let it grow. Let it guide you. And most importantly — **never stop learning**. Your intuition is always listening.