Choose Your Path: How to Use This Guide

> *"Not everyone learns the same way. Find your path below."*

This guide works for different people with different goals. **Don't read it cover to cover** unless that's your style. Instead, find yourself below and follow the recommended path.

🎯 Quick Self-Assessment

Answer these 3 questions to find your path:

1. What's your coding experience?

A) Complete Beginner

- Never written code before
- Don't know what "front-end" or "API" means
- Intimidated by technical terms

B) Dabbled Before

- Tried tutorials but never finished a project
- Understand some concepts but can't build alone
- Know what HTML/CSS are but not much else

C) Technical Non-Coder

- Work adjacent to developers (PM, designer, analyst)
- Understand architecture concepts
- Can read code but don't write it

D) Already Building with AI

- Built something with Cursor/Claude/ChatGPT
- Hit walls and need to understand what's happening
- Want to ship faster and more reliably

---

2. What's your primary goal?

1) Build a Specific Project

- Have an idea and want to make it real
- Need to ship something this month
- Don't care about theory, just want it working

2) Learn the Workflow

- Want to understand how AI building works
- Planning to build multiple projects
- Want foundational knowledge first

3) Solve Current Problem

- Stuck on authentication, deployment, or debugging

- Have a working project but something's broken
- Need specific answers now

4) Become Consistent Builder

- Want to ship weekly/monthly
- Looking to build a portfolio
- Interested in building as a skill, not just one project

---

3. How do you learn best?

α) Hands-On First

- Learn by doing, understand later
- Get frustrated reading without building
- Prefer following step-by-step tutorials

β) Comprehensive Top-Down

- Want to understand "why" before "how"
- Read documentation before coding
- Prefer systematic learning

γ) Just-In-Time Reference

- Look things up when you need them
- Skip theory, go straight to examples
- Bookmark reference guides

🗺️ Your Learning Path

Path 1: Complete Beginner (A + any goal)

If you answered A above, start here regardless of your goal:

```
1. QUICK-START.md (5 min) ← Get context
2. Chapter 4: Tutorial (1 hour) ← Build something immediately
3. Chapter 1: Method (15 min) ← Now understand what you just did
4. Chapter 2: Apps (20 min) ← Understand the architecture
5. Chapter 5: Building Blocks ← Reference as needed
6. Chapter 6: Debugging ← When things break
7. Chapter 7: Testing ← Before showing anyone
8. Chapter 8: What's Next ← Where to go from here
```

Why this order? You need proof you can do this BEFORE learning theory. Building first gives you context for everything else.

Time to first working app: 1 hour (Chapter 4)

Alternative if you're α (Hands-On First):

- Do Chapter 4 first
- Skip everything else until you hit a problem

- Use other chapters as reference when stuck
- Read Chapter 8 when ready for next project

Path 2: Dabbled Before (B + Goal 1 or 2)

You've seen tutorials but never shipped. This path focuses on finishing:

```
1. Chapter 1: Method (15 min) ← Understand the vibe coding approach
2. QUICK-START.md (5 min) ← Set up environment
3. Chapter 4: Tutorial (1 hour) ← Build end-to-end
4. Chapter 5: Building Blocks ← Reference when adding features
5. Chapter 7: Testing (20 min) ← Don't skip this
6. Chapter 6: Debugging ← Bookmark for when things break
7. Chapter 8: What's Next ← Plan your next build
```

Skip: Chapter 2 (you probably know this), Chapter 3 (use what tutorial uses)

Why this order? You have concepts but lack shipping experience. Focus on completing the tutorial, testing it, and deploying it. That's the skill you're missing.

Time to deployed app: 2 hours

Path 3: Technical Non-Coder (C + Goal 2 or 4)

You understand systems but haven't built. Fastest path to competence:

```
1. Chapter 1: Method (10 min) ← Skim - you get it
2. Chapter 3: Choosing Stack (15 min) ← Understand decisions
3. Chapter 4: Tutorial (45 min) ← You'll move faster than beginners
4. Chapter 5: Building Blocks ← Deep dive - this is your reference bible
5. Chapter 6: Debugging ← You'll need this often
6. Chapter 8: What's Next ← Build your roadmap
```

Skip: Chapter 2 (you know this), Chapter 7 (you'll figure it out)

Why this order? You have architecture knowledge. Focus on translation between "what you want" and "how to describe it to AI." Chapter 5 is your most valuable resource.

Time to competence: 3 hours reading + practice

Recommended: Read Chapter 5 twice. First time: overview. Second time: deep reference.

Path 4: Already Building with AI (D + any goal)

You've built something but hit limits. Fill in your gaps:

```
Start with your specific problem:

Stuck on authentication? → Chapter 5 (Authentication section)
Deployment failing? → Chapter 5 (Deployment section) + QUICK-START
Things breaking? → Chapter 6 (entire chapter)
```

```
Don't know if it's ready? → Chapter 7
Not sure what to build next? → Chapter 8
Want to understand architecture better? → Chapter 2
Choosing between frameworks? → Chapter 3
```

Don't read sequentially. Use this guide as a reference. Jump to what you need.

When to read full chapters:

- Chapter 6 (Debugging) - Read once, prevents hours of frustration
- Chapter 8 (What's Next) - Read when you finish your current project

Time to solve specific problem: 15-30 minutes

Path 5: Want to Build Specific Project (Any experience + Goal 1)

You have an idea. Here's the fastest path to live URL:

```
1. QUICK-START.md (5 min) ← Environment setup
2. Chapter 4: Tutorial (1 hour) ← Learn the pattern
3. Modify tutorial to your idea:
   - Keep the structure (Next.js, Supabase, Vercel)
   - Change the feature (bookmarks → your idea)
   - Deploy when core feature works
4. Chapter 5: Building Blocks ← Add features as needed
5. Chapter 7: Testing (15 min) ← Before announcing
```

Use as reference:

- Chapter 6 when stuck
- Chapter 5 for adding payments, email, file uploads

Skip: Chapter 1-2 (learn later), Chapter 8 (read after launching)

Why this order? Your goal is shipping. Learn minimum viable knowledge, ship fast, learn the rest later.

Time to deployed MVP: 4-8 hours over a weekend

Pro tip: Don't add features until core workflow is deployed. Ship early, iterate publicly.

Path 6: Want to Become Consistent Builder (C or D + Goal 4)

You want to ship weekly/monthly. Build the foundation:

```
Phase 1 (First Weekend):
1. Chapter 1: Method (15 min)
2. Chapter 4: Tutorial (1 hour)
3. Chapter 5: Building Blocks (1 hour) ← Study this
4. Ship the tutorial project

Phase 2 (Second Weekend):
1. Chapter 3: Choosing Stack (20 min)
2. Build micro-project with different stack
```

```
3. Chapter 6: Debugging (30 min)
4. Ship second project

Phase 3 (Third Weekend):
1. Chapter 8: What's Next (20 min)
2. Plan your building cadence
3. Set up templates from your first two projects
4. Ship third project (should take < 6 hours now)

Phase 4 (Ongoing):
- Reference Chapter 5 for new features
- Reference Chapter 6 when stuck
- Use Chapter 7 before each launch
- Reread Chapter 8 monthly for ideas
```

Why this order? Building consistency requires repetition. Ship three projects in three weeks. You'll learn patterns that make shipping fast.

Time to consistent shipping: 3 weeks

Key skill: By project 3, you should recognize patterns and reuse code. If not, slow down and study Chapter 5 more deeply.

📚 Learning Style Adaptations

If you're α (Hands-On First)

Your modified path (regardless of experience):

```
1. Chapter 4 → Build immediately
2. Get stuck? → Jump to relevant section in Chapter 5 or 6
3. Finish project? → Skim Chapter 7, deploy anyway
4. Deployed? → Read Chapter 8 for next project
5. Want to go deeper? → Read Chapter 1-3 after shipping
```

Your motto: "I'll understand it after I've built it."

Warning: You'll ship faster but may rebuild things. That's okay. Speed > perfect understanding.

If you're β (Comprehensive Top-Down)

Your modified path (regardless of experience):

```
1. README.md (10 min) ← Overview
2. Chapter 1: Method (15 min) ← Philosophy
3. Chapter 2: Apps (20 min) ← Architecture
4. Chapter 3: Stack (20 min) ← Decisions
5. GLOSSARY.md (10 min) ← Terminology
6. Chapter 4: Tutorial (1.5 hours) ← Now build with full context
7. Chapter 5-8 sequentially
```

Your motto: "I want to understand the system before I touch it."

Warning: You might over-learn before shipping. Force yourself to complete Chapter 4 even if you don't feel "ready."

If you're γ (Just-In-Time Reference)

Your modified path (regardless of experience):

```
1. Bookmark this guide
2. Start with Chapter 4 only
3. When you need something:
   - Authentication? → Chapter 5, Authentication section
   - Broken? → Chapter 6, search for your error type
   - New feature? → Chapter 5, relevant building block
   - Ready to ship? → Chapter 7 checklist
```

Your motto: "I'll read it when I need it."

Recommended bookmarks:

- Chapter 5 (Building Blocks) - Most referenced
- Chapter 6 (Debugging Checklist) - For when stuck
- Chapter 7 (Testing Checklist) - Before launch

Warning: You might miss foundational concepts. If you're rebuilding the same mistakes, spend 30 min reading Chapter 1.

🎓 Micro-Guides (Quick Reference)

Don't want to read full chapters? Use these micro-guides:

For Immediate Problems:

- `micro-guides/deploy-in-10-minutes.md` - Get your app live fast
- `micro-guides/fix-5-common-errors.md` - Debug the usual suspects
- `micro-guides/debugging-checklist.md` - Systematic troubleshooting

For Adding Features:

- `micro-guides/auth-quick-reference.md` - Add user accounts
- **Chapter 5: Building Blocks** - Payments, email, file uploads

For Shipping:

- `micro-guides/idea-to-live-url.md` - Complete shipping checklist
- **Chapter 7: Testing Checklist** - Pre-launch verification

When to use micro-guides: You have a specific question and need an answer in 5 minutes.

When to read full chapters: You keep hitting the same problems or want deep understanding.

🚀 Recommended First Steps by Path

If you haven't started yet:

1. Read this PATHWAYS.md (you're doing it!)
2. Find your path above
3. Do the first item in your path TODAY (even if just 15 minutes)
4. Schedule 1-2 hour block this week for Chapter 4

If you're midway through:

1. Find which path matches your experience + goal
2. See what you've already covered
3. Jump to the next recommended chapter
4. Skip what you've already mastered

If you finished the tutorial:

1. Read Chapter 7 (Testing)
2. Deploy your project (Chapter 5 → Deployment)
3. Read Chapter 8 (What's Next)
4. Plan your next build

If you're stuck:

1. Go directly to Chapter 6 (Debugging)
2. Use the debugging checklist
3. Find your specific error type
4. Come back to sequential learning after you're unstuck

🎯 Success Metrics by Path

How to know you're on the right path:

Path 1 (Complete Beginner):

- ✅ After 1 hour: You have something live on the internet
- ✅ After 1 week: You understand basic architecture
- ✅ After 2 weeks: You can explain how your app works
- ✅ After 1 month: You're building a second project

Path 2 (Dabbled Before):

- ✅ After 2 hours: You have a deployed project
- ✅ After 1 week: You're adding features to it
- ✅ After 2 weeks: You shipped a second project faster
- ✅ After 1 month: You understand patterns and can estimate build time

Path 3 (Technical Non-Coder):

- ✅ After 3 hours reading: You understand how to describe features to AI
- ✅ After first build: You see how architecture maps to prompts
- ✅ After 1 week: You're comfortable iterating with AI
- ✅ After 1 month: You're building without much reference to guide

Path 4 (Already Building):

- ✅ After 15 min: You found answer to your specific problem
- ✅ After 1 week: You're unstuck and shipping again
- ✅ After 2 weeks: You reference Chapter 5 regularly
- ✅ After 1 month: You're helping others debug their issues

Path 5 (Specific Project):

- ✅ After 1 hour: Tutorial completed
- ✅ After 1 weekend: Core feature of your idea working
- ✅ After 1 week: Deployed and getting feedback
- ✅ After 2 weeks: Iterating based on user feedback

Path 6 (Consistent Builder):

- ✅ After Week 1: First project shipped
- ✅ After Week 2: Second project shipped
- ✅ After Week 3: Third project shipped (in < 6 hours)
- ✅ After Month 2: You have templates and ship in < 4 hours
- ✅ After Month 3: You're teaching others

Not hitting these metrics? You might be on the wrong path. Reassess your answers above and try a different route.

🔄 When to Switch Paths

Switch from Comprehensive (β) to Hands-On (α) if:

- You've read 3+ chapters but haven't built anything
- You're overthinking instead of shipping
- You're waiting to "fully understand" before starting

Action: Force yourself to do Chapter 4 today.

Switch from Hands-On (α) to Comprehensive (β) if:

- You've built 2+ projects but keep hitting the same issues
- You're rebuilding features because you don't understand patterns
- Debugging takes longer than building

Action: Read Chapter 1-2, then skim Chapter 5 cover-to-cover.

Switch from Just-In-Time (γ) to Comprehensive (β) if:

- You've looked up the same thing 3+ times
- You feel lost about how pieces fit together
- Every new feature feels like starting from scratch

Action: Spend 1 hour reading Chapter 2 + Chapter 5 sequentially.

📖 Full Chapter Descriptions

If you want to know what's in each chapter before diving in:

Chapter 1: The Vibe Coding Method

- **What:** Philosophy and mindset of building with AI
- **Why read:** Understand the paradigm shift from traditional coding
- **Skip if:** You just want to build and don't care about theory
- **Read if:** You want to understand how this approach differs
- **Time:** 15 minutes

Chapter 2: Understanding the Shape of Apps

- **What:** How web apps are structured (front-end, back-end, database)
- **Why read:** Mental model for how pieces fit together
- **Skip if:** You're technical and know architecture
- **Read if:** You're confused about what AI is actually building
- **Time:** 20 minutes

Chapter 3: Choosing Your Stack

- **What:** Framework and tool decisions (Next.js vs Flask, Supabase vs Firebase)
- **Why read:** Understand trade-offs before committing
- **Skip if:** You're following the tutorial (uses Next.js + Supabase)
- **Read if:** Building second project or need different stack
- **Time:** 20 minutes

Chapter 4: Your First AI-Built App in One Hour

- **What:** Complete step-by-step tutorial (bookmark app with auth)
- **Why read:** Get hands-on experience building end-to-end
- **Skip if:** Never (everyone should do this)
- **Read if:** Always - this is the core of the guide
- **Time:** 1-1.5 hours

Chapter 5: Common Building Blocks

- **What:** Reference guide for adding features (auth, database, payments, email, file uploads, deployment)
- **Why read:** Most valuable reference chapter for adding features
- **Skip if:** Only building what's in tutorial
- **Read if:** Adding any new feature to any project
- **Time:** 1 hour to read, lifetime reference

Chapter 6: When Things Break

- **What:** Debugging process, common errors, how to get unstuck
- **Why read:** Save hours of frustration
- **Skip if:** Everything always works (it won't)
- **Read if:** Stuck on bug OR preventatively before you are
- **Time:** 30 minutes to read, save hours when needed

Chapter 7: How to Know It Actually Works

- **What:** Testing checklist, device testing, pre-launch verification
- **Why read:** Ship with confidence, avoid embarrassing bugs
- **Skip if:** Building only for yourself
- **Read if:** Showing to users or launching publicly
- **Time:** 20 minutes

Chapter 8: What's Next

- **What:** Next project ideas, building consistently, community
- **Why read:** Don't stop at one project
- **Skip if:** Haven't finished first project yet
- **Read if:** Shipped something and ready for more
- **Time:** 15 minutes

🆘 Still Not Sure? Start Here:

Default path for maximum people:

```
Day 1 (1 hour):
- QUICK-START.md (5 min)
- Chapter 4 up to "Deploy Your App" (55 min)

Day 2 (1 hour):
- Finish Chapter 4 deployment
- Add one custom feature of your choice

Day 3 (30 min):
- Chapter 7 testing checklist
- Show to 2 friends, get feedback

Day 4 onwards:
- Read Chapter 8
- Plan next project
- Reference other chapters as needed
```

By end of week 1, you'll have:

- Deployed working app
- Understanding of the build process
- Confidence to build next project

That's the goal. Start there, adjust as you learn your style.

📞 Need Help Choosing?

If you're still unsure which path to take, ask yourself:

"Do I have a deployed working app?"

- ❌ No → Do Chapter 4 today
- ✅ Yes → Go to Chapter 8, plan next build

"Am I stuck on a specific problem right now?"

- ✅ Yes → Chapter 6 (Debugging) immediately
- ❌ No → Follow one of the paths above

"Do I learn better by reading or doing?"

- Doing → Hands-On (α) adaptation
- Reading → Comprehensive (β) adaptation
- Both → Follow standard path for your experience level

Remember: The best path is the one you'll actually follow. Don't overthink it. Pick one, start today, adjust as you go.

You're not locked in. These are guides, not rules. Mix and match as needed.

The only wrong choice is not starting.

Now pick your path and go build something.

Connect & Share

💌 **Newsletter**: Build to Launch - Weekly AI building tips, templates, and real builder stories

✍️ **Medium**: AI Builders - Read more articles and guides

💬 **Reddit**: r/VibeCodingBuilders - Join the community

🦋 **Bluesky**: @jenny-ouyang - Connect

💼 **LinkedIn**: Jenny Ouyang - Connect