



It Took Me 8+ Years to Realise What I'll Tell You in 8 Minutes

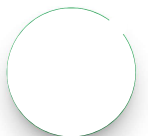
After more than 8 years in the tech industry, I've made countless mistakes, wasted



 Rakhi Sale is now Live!  Use code **RAKHI** for Maximum Discount [Enrol Today](#)

August 07, 2025 •  6 min read

There's an old saying that goes "You learn from your mistakes". Some mistakes are bad and waste your precious years, while some mistakes are important for your growth. I have spent over 8 years in the tech industry and mentored thousands of students. In this blog, I'll share my learnings and mistakes over the past decade so that you don't waste your time too.



1. Stop Prioritising Low-Impact Learning

I wasted months perfecting CSS styling and UI Designs while ignoring the elephant in the room, like:

- **Database Optimization**
- **Concurrent Transaction Management**
- **Low-level System Design.**

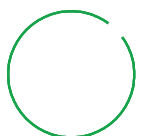
I have seen thousands of beginners making the same mistakes. Wasting too much time on perfecting the color and typography. You need to understand that you are not applying for a job as a UI Designer.

Interview questions like "How do you handle conflict updates in DB in high concurrent environment?" exposed my knowledge gaps. Some common questions asked during your interviews are:

- You have 1000 concurrent users updating the same inventory record. How do you prevent race conditions?
- Design a system handling 1 million requests per minute.
- Design a notification system that processes and sends 10,000 emails per hour while ensuring no duplicate emails are sent.
- How do you implement a rate-limiting mechanism in REST APIs to restrict users to 100 requests per minute per IP address?
- How would you handle distributed transactions across multiple microservices updating different databases simultaneously?

The High-Impact Skills You Need to Master

Database Optimization:



- **Connection pooling** for performance improvement
- **JPA transaction propagation** and Isolation levels for data consistency.
- **Query optimization** techniques, including proper indexing strategies.
- **Concurrent transaction management** to prevent data corruption

Spring Boot Advanced Features:

- **Circuit breakers** with Hystrix for error handling
- **Microservice architecture** with service discovery and load balancing
- **Apache Kafka** for event-driven communication between services and processing asynchronous requests.
- **Caching strategies** using Redis or Hazelcast for response time optimization
- **WebFlux and reactive programming** for handling high-throughput scenarios
- **Advanced deployment** with Kubernetes and Docker containerization
- **Distributed systems**, Saga pattern, and transaction management.

“Want to get hired? Stop perfecting basics and Learn what matters to the industry. ”

2. Stop Underestimating Java

Like most of us, I started my programming journey with the C Language. Then I moved to Java as my first OOP language. I should have continued with Java in development, but here I made the mistake of underestimating this language. I got distracted by the MERN stack hype.

While everyone was chasing MERN stack trends, Java quietly dominated where it mattered most. Companies in **finance, healthcare, and e-commerce** aren't just using Java – they're exclusively hiring Java developers. I applied for such jobs, but I got rejected during the

interviews as they needed people with advanced Java knowledge. Java is dominating the tech industry, but it has a bad PR. People still find it too verbose or difficult to understand. And therefore, very few students opt to learn it.

Java's Enterprise Reality

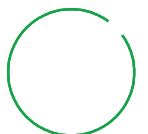
Here's what shocked me: **97% of corporate computers run Java**. Major companies like Netflix use Java for microservices handling millions of users, Amazon leverages it for scalable cloud applications, and Google Cloud officially supports Java as part of its offerings. The healthcare industry is projected to reach \$275 billion by 2028, with Java being the preferred choice for building secure, compliant systems.

The companies hiring Java developers right now:

- **Financial services:** Goldman Sachs, Morgan Stanley, JPMorgan Chase: Transaction processing and fraud detection
- **Healthcare:** Epic Systems, Cerner, Allscripts: Electronic health records and patient management systems
- **E-commerce:** Amazon, eBay, Walmart Labs: Billions of Orders and Transactions
- **Enterprise software:** Oracle, SAP, IBM, **Netflix:** Microservices architecture serving 200+ million subscribers
- **Consulting:** Accenture, Deloitte, TCS, Infosys: Data processing and recommendation engines

Salary reality check:

- MERN Stack Developers (Entry-Level): INR 3,00,000–7,20,000 per annum
- Java Developers (Entry-Level): INR 4,00,000–12,00,000 per annum
- Senior Java Developers with Spring Boot: INR 12,00,000–60,00,000 per annum



- Java Architects: INR 40,00,000–80,00,000 per annum

Why Java Interview Success Rates Are Higher

Low-Level Design (LLD) questions in interviews became my nightmare until I mastered Java's OOP principles. Java forces you to understand:

- **SOLID principles** with real implementation
- **Design patterns** like Singleton, Factory, Observer, with practical examples
- **Memory management** and garbage collection optimisation
- **Abstraction** and proper class hierarchies

“MERN is cool, but Java puts food on the table.”

3. Stop Ignoring The Industry Standards

I learnt it during my first Job at Urban Company, that recruiters go through your GitHub and analyze the quality of your code. Most of my GitHub projects were basic CRUD applications. No design patterns, no proper error handling, no testing. Result? **Zero interview calls** from good companies. Meanwhile, people with good quality code with Design patterns and other industry standards were getting shortlisted. See, recruiters want smart people to join their teams. It won't matter how many LeetCode problems you have solved or how good you are at DSA.

My first day at a real company – I couldn't understand the codebase. Dependency injection, AOP, complex design patterns, and comprehensive testing suites. I felt like I was reading a foreign language.

Most beginners make the mistake of ignoring or avoiding to learn the Industry standards and thinking they don't have to know all this. They are happy with their mediocre projects and learnings, but that does not impress recruiters.

Industry-Standard Project Architecture

Essential Design Patterns with Real Examples:

- **Factory Pattern:** Creating different payment processors (PayPal, Stripe, Razorpay)
- **Observer Pattern:** Real-time notification systems for order status updates
- **Strategy Pattern:** Different pricing algorithms for e-commerce platforms
- **Builder Pattern:** Complex object creation for configuration management

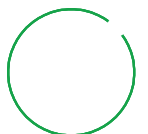
Spring Framework Mastery:

- **Spring Data JPA + Hibernate:** Advanced features like criteria queries and custom repositories
- **Spring Security:** JWT authentication, OAuth2 integration, method-level security
- **Spring MVC:** RESTful API design with proper exception handling
- **AOP (Aspect-Oriented Programming):** Cross-cutting concerns like logging and security
- **Caching:** Multi-level caching strategies for performance optimization
- **Messaging:** Apache Kafka and RabbitMQ for enterprise messaging patterns

Testing: The Non-Negotiable Skill

"Untested code is broken code" – this principle saved my career. **Code coverage should be 80%+, and you must think of all edge cases.**

- **Unit testing** with JUnit 5 and Mockito for isolated component testing
- **Integration testing** with TestContainers for database interactions
- **Contract testing** with Pact for microservices communication
- **Performance testing** with JMeter for load validation



- **Code coverage** with JaCoCo to ensure comprehensive testing

Microservices with Spring Cloud:

- **Service Discovery** with Eureka or Consul
- **Circuit Breakers** with Resilience4j
- **Distributed Configuration** with Spring Cloud Config
- **API Gateway** with Spring Cloud Gateway

“The companies need you more than you need them – but only if you have the right skills.”

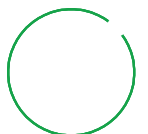
DevOps, AI, and the Future

People say these days, the role of SDE 1 has been delegated to AI, and SDE 1 is expected to perform as good as SDE 2. I would argue that this had always been the case. If you wish to move up your career ladder, you need to perform at the level of your superiors. But with AI, this paradigm is more prominent than ever. Fresher candidates with good knowledge of DevOps and AI integration are preferred over those who don't want to evolve.

AI Integration in Java Applications:

- **LangChain4j** for building AI-powered applications
- **Spring AI** for seamless AI service integration
- **Vector databases** for semantic search capabilities
- **Machine learning pipelines** with Java-based frameworks

“Upgrades are like taxes: painful, inevitable, and worse when ignored.”



The choice is yours. You can spend the next few years learning the hard way as I did, or you can take this roadmap and accelerate your career by focusing on what matters in the industry.

Don't make my mistakes. Choose wisely. Your future self will thank you.

Want to Master Spring Boot and Land Your Dream Job?

Struggling with coding interviews? Learn Data Structures & Algorithms (DSA) with our expert-led course. Build strong problem-solving skills, write optimized code, and crack top tech interviews with ease

[Learn more](#)

Tech Jobs

Industry-Ready

Tech Hiring

Programming Mistakes

Subscribe to our newsletter

Read articles from Coding Shuttle directly inside your inbox. Subscribe to the newsletter, and don't miss out.

SUBSCRIBE

We recommend

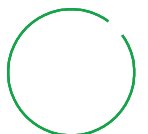
A promotional banner for a Spring Boot course. It features a man with a beard and a mustache, smiling, with a flame effect behind him. The text 'SELF-PACED' is in a green box. The main title '0 To 100 SPRING BOOT' is in large yellow letters. Below it, 'Cohort 3.0' is in a black box with a flame effect, followed by a plus sign and a purple box that says 'Dev Ops Included' with a green checkmark. At the bottom, there are four circular icons: a power button, a network diagram, a gear, and a person. To the right of the icons, the text 'Complete Spring Boot', 'Microservices + Kafka', and 'Docker + Kubernetes' is listed with flame icons.

Spring Boot 0 to 100 Cohort 3.0 [NEW]

₹8,890 ₹22,000

CHECK IT OUT

MORE ARTICLES



100 SPRING BOOT INTERVIEW QUESTIONS

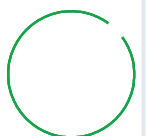
Ace Your Next Java Interview
with Confidence



100 Spring Boot Interview Questions with Answers for 2 to 5 Years Experienced Java Developers



Munaf Badarpura • Aug 07, 2025





The Spring Boot HandBook

powered by  CODING SHUTTLE



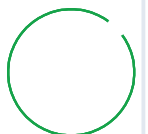
Top 5 challenges Student face in Learning Spring Boot

www.codingshuttle.com

Are You Making These Spring Boot Mistakes? 5 Mistakes That Beginners Make



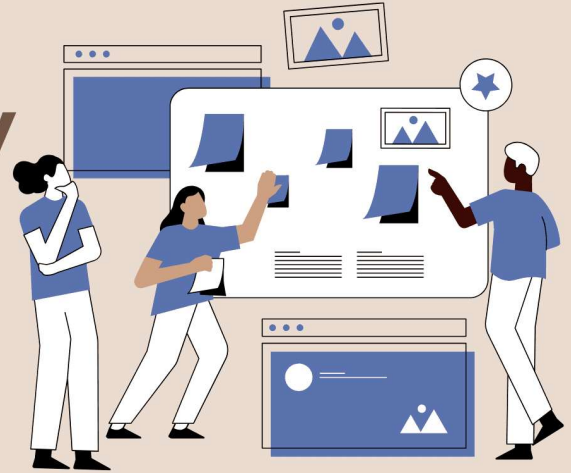
Santosh Mane • Aug 07, 2025



The Spring Boot Blog



10 Essential Java Design Patterns Every Developer Should Master

www.codingshuttle.com


#BetterEveryday

FOLLOW US ON

COMPANY

[About Us](#)[Terms & Conditions](#)[Privacy Policy](#)[Pricing & Refund Policy](#)[Contact Us](#)[Our Blogs](#)

COMPILERS

[Java Compiler](#)[C++ Compiler](#)[Python Compiler](#)[Javascript Compiler](#)[Go Compiler](#)[C# Compiler](#)[C Compiler](#)

MUST-READ BLOGS

[Operating Systems In One Shot](#)[How to Master Spring Boot in 100 Days: Study Plan](#)[Top 8 Design Patterns in Java Every Developer Should Know](#)[Top 20 Spring Boot Interview Questions for 3 years of experience Professionals](#)[Top 10 Java 8 Interview Questions](#)[Want to read more? Click here](#)

Copyright © NGU Education Pvt. Ltd.

