

# hi Dia

Yesss. That's the move. Not random learning — structured growth. I like this version of you 😊🔥

We'll build a **practical backend roadmap** assuming ~1-2 hours per day. No fluff. Only what makes you dangerous inside **Accenture**.

---



## 12-Week Backend Developer Roadmap



### Phase 1 (Weeks 1–3): Core Java Mastery

🎯 Goal: Think like a backend engineer

#### Week 1:

- OOP principles (deep understanding, not definition level)
- SOLID principles
- Access modifiers
- Abstraction vs Interface (real use cases)

#### Week 2:

- Collections (List, Set, Map, HashMap internals)
- equals() & hashCode()
- Comparable vs Comparator
- Generics

#### Week 3:

- Exception handling best practices
- Multithreading basics (Thread, Runnable, Executor)
- Synchronization
- Java Memory basics (high level)

👉 Build small console projects:

- Bank system
- Inventory manager
- Multithreaded counter

## Phase 2 (Weeks 4–7): Spring Boot Deep Dive

We now move to real enterprise backend using  
Spring Boot

---

### **Week 4:**

- What is Spring & Dependency Injection
- Create REST APIs
- `@RestController`
- `@Service`
- `@Repository`
- Postman testing

Build: Simple CRUD app.

---

### **Week 5:**

- Connect to MySQL
- JPA & Hibernate
- Entity relationships (OneToMany, ManyToOne)
- DTO pattern

Build: Student management system.

---

### **Week 6:**

- Exception handling (global)
- Validation
- Logging
- API response standardization

Make it production-like.

---

### **Week 7:**

- JWT Authentication

- Basic role-based access
- Password hashing
- API security basics

Now you're entering serious backend territory.

---

## Phase 3 (Weeks 8–9): SQL Mastery

### Week 8:

- Advanced joins
- Subqueries
- Indexing
- Query optimization

Practice on:

- LeetCode SQL
  - HackerRank SQL
- 

### Week 9:

- Transactions
- ACID
- Isolation levels
- Stored procedures
- Database design (normalization)

Backend dev without SQL depth = incomplete.

---

## Phase 4 (Weeks 10–11): Git + Project Discipline

### Week 10:

- Branching strategy
- Pull requests
- Merge conflicts
- Rebase vs merge

Simulate team workflow.

---

## Week 11:

- Project structure best practices
  - Logging frameworks
  - Clean code principles
  - Basic system design (monolith vs microservices idea level)
- 

## Phase 5 (Week 12): Deployment Awareness

You don't need mastery. Just awareness.

- What is Docker
- What is CI/CD
- What is AWS EC2
- How deployment works

Understand how your backend reaches users.

---

## Bonus: Corporate Skill (Do this weekly)

- Practice explaining your project in 3 minutes
- Record yourself explaining OOP
- Learn to give structured updates

Technical skill + communication = growth.

---

Now let me ask something important.

Are you planning to start this immediately?

And realistically, how many hours per day can you give?

Because I can optimize this roadmap for:

- 1 hour/day mode
- 2 hour/day aggressive mode
- Weekend heavy mode

Tell me your realistic commitment. No overconfidence. 😊

Now that's what I like to hear 😊🔥

2 hours daily is PERFECT. Not burnout. Not lazy. Sustainable domination.

Okay, we'll structure this properly. No random YouTube hopping. Clear daily breakdown.

This is your **2-Hour Backend Growth System (12 Weeks)** before joining **Accenture**.

---

## Daily Structure (2 Hours)

 First 60 mins → Concept Learning

 Next 60 mins → Implementation / Coding Practice

No passive learning allowed.

---

## **17** Weeks 1–3: Core Java Power

**Daily Split Example:**

**Hour 1**

- OOP / SOLID / Collections / Threads (depending on week)

**Hour 2**

- Build small console apps
- Rewrite examples without looking
- Solve 1–2 Java logic problems

 Rule: If you cannot explain it in simple language, you don't know it yet.

---

## **17** Weeks 4–7: Spring Boot Builder Mode

Working with

**Spring Boot**

---

**Hour 1**

- Learn Spring concepts (DI, REST, JPA, Security)

## Hour 2

- Build project step-by-step

By end of Week 7 you should have:

- One proper CRUD backend project
- One secured JWT-based API
- Clean folder structure

This becomes your confidence booster.

---

July  
17

## Weeks 8–9: SQL Domination

### Hour 1

- Advanced queries
- Transactions
- Indexing

### Hour 2

- Practice queries on real datasets
- Try optimizing slow queries

Backend dev who knows SQL well = respected.

---

July  
17

## Weeks 10–11: Git + Clean Code

### Hour 1

- Git workflows
- Branching
- Rebase vs Merge
- Pull requests

### Hour 2

- Refactor your Spring project
- Improve logging
- Add proper exception handling

Turn your project from “college level” to “enterprise level”.

---

July  
17

## Week 12: Deployment Awareness

Learn:

- What is Docker
- CI/CD basics
- How backend is deployed
- Basic AWS idea

No deep dive. Just conceptual clarity.

---



### Important Mindset Rule

You are not studying to pass exams anymore.

You are studying to:

- Be reliable in meetings
- Understand tickets quickly
- Fix bugs without panic
- Speak confidently in standups

That's real growth.

---

Now one honest question:

Are you disciplined enough to do 2 hours daily even when motivation drops?

Because consistency will make you dangerous here.

If you say yes, I'll make you a Week 1 micro-plan starting tomorrow ☀️🔥

That's the energy 😊🔥

Alright. No more theory talk. We start executing.

This is your **Week 1 Micro Plan (Backend Domination Begins)**.

Focus: **Core OOP + Clean Coding Foundation**

---

## ◆ Day 1 – OOP Deep Understanding

### Hour 1 (Concept)

- What is OOP really?
- Encapsulation (real-world examples)
- Abstraction
- Interface vs Abstract class (WHEN to use each)

### Hour 2 (Code)

Build:

- A simple `BankAccount` class
- Add deposit, withdraw methods
- Hide internal balance properly

Goal: Write clean, modular code.

---

## ◆ Day 2 – Inheritance & Polymorphism

### Hour 1

- Method overloading vs overriding
- Runtime vs compile-time polymorphism
- `super` keyword

### Hour 2

Build:

- A `Shape` base class
- `Circle`, `Rectangle` subclasses
- Override `area()` method

Understand dynamic binding properly.

---

## ◆ Day 3 – SOLID Principles (Very Important)

### Hour 1

- Single Responsibility

- Open/Closed
- Liskov Substitution

Don't memorize. Understand WHY they exist.

## Hour 2

Refactor your previous code using SOLID principles.

Enterprise coding = maintainable coding.

---

## ◆ Day 4 – Collections Deep Dive

### Hour 1

- List vs Set vs Map
- HashMap internal working (important)
- equals() & hashCode()

### Hour 2

Build:

- Student record system using HashMap
- Practice sorting using Comparator

Backend engineers must understand data structures deeply.

---

## ◆ Day 5 – Exception Handling

### Hour 1

- Checked vs unchecked
- Custom exceptions
- Best practices

### Hour 2

Add:

- Proper exception handling to Bank system
- Create custom InsufficientBalanceException

Clean error handling = professional code.

---

## ◆ Day 6 – Multithreading Basics

### Hour 1

- Thread vs Runnable
- ExecutorService
- Synchronization

### Hour 2

Build:

- A simple multithreaded counter
- Try to see race condition
- Fix it using synchronized

This will separate you from average freshers.

---

## ◆ Day 7 – Revision + Explanation Day

This is critical.

### Hour 1

Revise everything.

### Hour 2

Explain OOP, SOLID, Collections aloud like you're in an interview.

If you can explain clearly, you're growing.

---

## 🔥 Rule for This Week

No skipping.

No passive watching.

No "I'll do tomorrow."

2 hours daily. Even if tired.

---

Next week we move into serious territory toward  
Spring Boot

You're not preparing to survive Accenture.

You're preparing to walk in confident.

Tomorrow is Day 1.

Message me after finishing Day 1. I'll test you a little 😊