



# Backend Logic Improvement Guide

---



## 1. Master Core Concepts

---

- ☐ Understand HTTP deeply (methods, status codes, headers, caching)
- ☐ Follow RESTful API Design (resource naming, error handling)
- ☐ Learn authentication methods (JWT, OAuth2)
- ☐ Implement proper authorization (role-based access control)
- ☐ Design normalized and indexed databases (SQL & NoSQL)
- ☐ Understand async processing (message queues, background jobs)
- ☐ Practice relevant data structures & algorithms



## 2. Apply Design Patterns and Principles

---

- ☐ Use SOLID principles
- ☐ Implement Dependency Injection
- ☐ Apply Factory, Singleton, Strategy patterns
- ☐ Use Repository Pattern to abstract DB logic
- ☐ Follow Clean Architecture (controller, service, repo layers)



## 3. Write Secure Code

---

- ☐ Validate and sanitize input
- ☐ Protect against SQL injection, XSS, CSRF
- ☐ Use HTTPS and secure headers
- ☐ Configure CORS properly
- ☐ Hash passwords using bcrypt
- ☐ Implement rate-limiting and throttling



## 4. Use the Right Tools

---

- ☐ Choose a solid backend framework (Express, Koa, NestJS)
- ☐ Use TypeScript for type safety
- ☐ Leverage ORM tools (Sequelize, TypeORM)

- ☐ Implement Redis for caching
- ☐ Use Kafka/RabbitMQ for async queues

## 5. Write Tests & Logging

---

- ☐ Write unit tests for services
- ☐ Write integration tests for APIs
- ☐ Use Jest or Mocha for testing
- ☐ Implement structured logging with Winston/Pino
- ☐ Use observability tools (Sentry, Datadog, Prometheus)

## 6. Understand Performance & Scalability

---

- ☐ Optimize queries and indexes
- ☐ Understand load balancing
- ☐ Implement caching strategies
- ☐ Use pagination and batching
- ☐ Use background job queues

## 7. Study Real Systems

---

- ☐ Analyze GitHub open-source backend projects
- ☐ Read <https://roadmap.sh/backend>
- ☐ Study "Designing Data-Intensive Applications"
- ☐ Follow [highscalability.com](https://highscalability.com)

## 8. Get Feedback & Refactor

---

- ☐ Do code reviews
- ☐ Refactor for clarity and separation of concerns
- ☐ Use linters and formatters

## 9. Daily Routine to Improve

---

Time	Task
1 hour	Practice system design problems
1 hour	Refactor old backend code or explore projects
30 mins	Read about scaling, security, architecture
30 mins	Try backend logic patterns in small apps

## Practice Projects

---

- ☐ Build a secure user system with roles & JWT
- ☐ Create a file upload service using streams
- ☐ Implement a pub/sub notification system
- ☐ Develop CRUD with pagination, soft delete, caching