50+ Technical Terms in Programming

Core Software Concepts

Simplicity: Keep it easy and clean. No mess.
Flexibility: Can change easily if needed.
Redundancy: Backup plans. Just in case.
Modularity: Break your code into neat, small parts.
Reusability: Write once, use many times.
Scalability: Grows easily without crashing.
Maintainability: Easy to fix or update.
Portability: Runs on any system (Windows, Linux, Mac).
Reliability: Works properly. All the time.
Availability: Always ready to use.
Efficiency: Uses less power and memory.
Performance: Fast and smooth.
Security: Safe from hackers and attacks.
Integrity: Data is correct and untouched.
Confidentiality: Data stays private.
Abstraction: Show only what’s important.
Encapsulation: Wrap code + data together. Keep it safe.
Polymorphism: One thing, many forms.
Inheritance: Child code gets stuff from parent code.
Overloading: Same name, different input.
Overriding: Child changes how parent behaves.

System & Network Terms

Latency: Delay. Like buffering in a slow video.
Bandwidth: Data speed limit.
Throughput: How much data actually gets through.
Load Balancing: Share work across servers.
Fault Tolerance: Still works even if parts break.
Concurrency: Tasks running at the same time.
Parallelism: Doing many things at once.
Deadlock: Two programs waiting for each other forever.
Race Condition: Programs fighting over who goes first.
Thread: A lightweight task.
Process: A running program.

Database Terms

Normalization: Organize data to avoid copies.
Denormalization: Add some copies for faster speed.
Index: Like a table of contents for quick search.
Transaction: All steps succeed or none do.
Atomicity: All-or-nothing.
Consistency: Data stays correct.
Isolation: One operation doesn’t mess up another.
Durability: Data survives even after a crash.
Primary Key: Unique ID for each row.
Foreign Key: Link between tables.
Schema: The blueprint of your database.
Query: Ask for data.

Programming Principles & Practices

DRY: Don’t repeat yourself.
KISS: Keep it simple, stupid.
YAGNI: You ain’t gonna need it.
SOLID: 5 rules for better object-oriented code.
Tight Coupling: Too much dependency — risky.
Loose Coupling: Less dependency — flexible.
Cohesion: A module should do one thing well.
Code Smell: Something’s off with the code — fix it.
Refactoring: Clean up code without changing what it does.
Debugging: Fixing errors.

Unit Testing: Test small pieces of your app.
Mocking: Fake parts to test real parts.