CONCEPTS EVERY DEVELOPER SHOULD KNOW

💡 Concept	💬 What It Means	✅ Why It's Useful
Big-O Notation	Measures time/space complexity	Write faster, scalable code
Hashing	Key-value storage via hash functions	Basis for maps, databases, caching
Recursion	Function calls itself to break down problems	Solves trees, graphs, complex logic
Call Stack	Tracks function calls (LIFO)	Debug errors, prevent overflows
Memory Leaks	Forgotten memory still in use	Prevents crashes & slow apps
Event Loop	JS async processing queue + stack	Understand async behavior in web apps
Promises	Manage async with .then() / async-await	Avoid callback hell in JavaScript
Data Structures	Arrays, Sets, Maps, Trees, Linked Lists	Foundation of any application logic
Algorithms	Sorting, searching, greedy, backtracking	Core to problem-solving + interviews
Networking Basics	HTTP, DNS, REST, TCP/UDP	Needed to build or debug any online app
Authentication	JWT, OAuth, Sessions	Required in every full-stack app
Databases	SQL vs NoSQL, joins, indexes	All real apps use one — must understand logic
State Management	Store/update data in frontend/backend	Crucial for React, Vue, Angular, etc.
Concurrency	Multiple processes at once (threads, async, etc.)	Helps with performance, backend logic
Error Handling	Try/catch, graceful fails, logging	Builds robust and safe apps