

Profile 1 — Thinking: General

Prompt 1 — Energy Transition Strategy

A country currently generates 70 percent of its electricity from coal and natural gas and wants to reach 90 percent renewable electricity within 15 years. Develop a detailed transition strategy. Compare solar, wind, hydroelectric, geothermal, nuclear, battery storage, pumped hydro, and demand-response technologies. Analyze technical feasibility, grid stability, transmission requirements, seasonal energy storage, land use, supply-chain constraints, workforce requirements, capital costs, operating costs, energy security, and environmental impacts. Identify the major trade-offs and failure modes. Then propose a phased 15-year implementation plan with measurable milestones. Explain the reasoning behind your recommendations in detail.

Prompt 2 — Urban Growth Strategy

Design a comprehensive strategy for a rapidly growing city of five million people that expects its population to increase by 40 percent over the next twenty years. The city already faces traffic congestion, housing shortages, aging water systems, electricity demand growth, air pollution, and limited public funds. Develop an integrated plan covering transportation, housing, water, electricity, waste management, public health, climate resilience, and digital infrastructure. For each area, analyze competing priorities, dependencies, risks, costs, and possible unintended consequences. Explain how the city should sequence investments over twenty years and how success should be measured.

Prompt 3 — Industrial Decarbonization

A manufacturing company operates twelve factories across multiple countries and wants to reduce its total carbon emissions by 80 percent while keeping production costs competitive. Build a detailed decision framework for achieving this target. Consider renewable electricity, electrification of industrial processes, energy efficiency, alternative fuels, supply-chain changes, transportation, carbon capture, equipment replacement cycles, and carbon offsets. Discuss financial, technical, regulatory, operational, and supply-chain risks. Compare short-term and long-term strategies and propose a prioritized implementation roadmap with clear reasoning.

Prompt 4 — Water Security

Imagine that a region experiences a severe five-year drought while its population and agricultural demand continue to grow. Develop a detailed water-security strategy that balances urban consumption, agriculture, industry, ecosystem protection, and future

population growth. Evaluate desalination, wastewater recycling, groundwater extraction, reservoirs, conservation, smart metering, agricultural efficiency, water pricing, and inter-regional transfers. Analyze economic, environmental, technical, political, and social trade-offs. Construct a phased response for immediate emergency measures, medium-term stabilization, and long-term resilience.

Prompt 5 — Resilient Digital Infrastructure

A national government wants to build a resilient digital infrastructure strategy for the next fifteen years. It must support cloud computing, artificial intelligence, healthcare, banking, government services, education, and emergency communications while remaining resilient to cyberattacks, natural disasters, supply-chain disruptions, and power failures. Develop a detailed architecture and policy strategy. Compare centralized and distributed infrastructure, public cloud, sovereign cloud, edge computing, local data centers, backup power, network redundancy, cybersecurity controls, and disaster recovery. Identify trade-offs and propose measurable resilience targets.

Prompt 6 — AI, Automation and the Workforce

Analyze how a country should prepare its economy and workforce for widespread adoption of artificial intelligence and automation over the next twenty years. Consider productivity, job displacement, creation of new occupations, education, reskilling, inequality, taxation, social programs, small businesses, research investment, cybersecurity, energy demand, and international competitiveness. Present multiple plausible future scenarios rather than assuming a single outcome. For each scenario, identify major risks, leading indicators, and policy responses. Finish with a staged strategy that can adapt as conditions change.

Profile 2 — Thinking: Precise Coding

Prompt 1 — Longest Consecutive Sequence

Design and implement in TypeScript an $O(n)$ algorithm that accepts an unsorted array of integers and returns: the length of the longest consecutive sequence, the sequence itself, and the starting and ending values. Duplicate values may appear and must not affect the result. If multiple sequences have the same maximum length, return the one with the smallest starting value. Do not sort the input. Explain the algorithm, prove why its expected time complexity is $O(n)$, analyze space complexity, discuss edge cases, and provide clean production-quality TypeScript code followed by at least eight meaningful test cases.

Prompt 2 — LRU Cache

Design a production-quality TypeScript implementation of an LRU cache with $O(1)$ average-time get and put operations. Do not use JavaScript Map insertion order as the eviction mechanism. Implement the required hash-map and doubly-linked-list logic explicitly. Support updating existing keys, capacity one, arbitrary string keys, and generic values. Explain every major design decision, identify invariants that must remain true, analyze time and space complexity, discuss common implementation bugs, and provide a comprehensive test suite covering eviction, updates, repeated access, and edge cases.

Prompt 3 — Dependency-Aware Task Scheduler

You are designing a task scheduler in TypeScript. Each task has an ID, execution duration, and zero or more dependency IDs. Design an algorithm that validates the dependency graph, detects cycles, produces a valid execution order when possible, and calculates the earliest possible completion time assuming an unlimited number of workers and that a task can start only after all dependencies finish. Explain the graph representation, cycle-detection method, topological ordering, dynamic-programming step, complexity, and failure cases. Then provide complete TypeScript code and a set of nontrivial tests.

Prompt 4 — High-Volume Job Processing

Design a thread-safe job-processing architecture for a service that receives up to 50,000 jobs per minute. Jobs may be retried, workers may crash, duplicate messages may be delivered, and jobs must never be considered successfully completed more than once from the application's perspective. Explain how you would implement idempotency, retries with exponential backoff, dead-letter handling, leases or visibility timeouts, concurrency control, database transactions, observability, graceful shutdown, and recovery after failure. Then provide detailed TypeScript pseudocode or implementation-level code for the core worker loop and explain race conditions that must be prevented.

Prompt 5 — Large-Scale Interval Processing

Implement in TypeScript a function that receives millions of time intervals represented as start and end integer timestamps and returns the merged non-overlapping intervals, total covered duration, largest uncovered gap between the minimum and maximum timestamps, and maximum number of simultaneously active intervals. Clearly define how touching intervals and zero-length intervals are handled. Develop an efficient algorithm, analyze its time and space complexity, explain alternative approaches for

very large datasets that do not fit into memory, and provide production-quality code with extensive edge-case tests.

Prompt 6 — Node.js Performance Investigation

A Node.js API has gradually become slow under load. Requests perform authentication, query PostgreSQL, call two external services, transform large JSON objects, and publish an event. At 50 requests per second latency is 120 ms, but at 500 requests per second p95 latency exceeds 8 seconds and memory continually increases. Develop a systematic debugging and optimization plan. Reason about event-loop blocking, connection pools, unbounded concurrency, memory leaks, garbage collection, database indexes, N+1 queries, HTTP connection reuse, timeouts, retries, queues, caching, backpressure, and instrumentation. Include concrete TypeScript/Node.js examples showing how you would restructure the most likely bottlenecks.

Profile 3 — Instruct: General

Prompt 1 — Modern Electrical Power Grids

Write a detailed educational guide explaining how modern electrical power grids work for a reader with no engineering background. Cover electricity generation, transformers, high-voltage transmission, substations, distribution networks, frequency and voltage control, supply-demand balancing, grid operators, renewable-energy integration, battery storage, blackouts, and grid restoration. Organize the answer with clear headings, use practical examples and analogies, define technical terms when first introduced, and finish with a concise summary of the entire electricity journey from power plant to household.

Prompt 2 — How a Web Application Works

Create a comprehensive beginner-friendly explanation of how a web application works from the moment a user enters a URL until the requested page appears. Explain DNS, IP addresses, TCP, TLS, HTTP, browsers, CDNs, load balancers, reverse proxies, web servers, APIs, application servers, databases, caches, cookies, authentication, JavaScript execution, rendering, and browser storage. Follow the request chronologically and clearly explain which component is responsible for each stage. Include a realistic example involving a user signing into an online store.

Prompt 3 — Relational vs. Document Databases

Write a structured technical guide comparing relational databases and document databases. Explain schemas, tables, documents, relationships, normalization,

denormalization, indexes, transactions, ACID properties, consistency, horizontal and vertical scaling, replication, sharding, query patterns, data integrity, and operational complexity. Use PostgreSQL and MongoDB as concrete examples without presenting either as universally superior. Include several application scenarios and explain which characteristics should influence the database choice in each scenario.

Prompt 4 — Machine Learning Lifecycle

Explain the complete machine-learning lifecycle to a software developer who understands programming but has never built a machine-learning system. Cover problem definition, data collection, cleaning, labeling, exploratory analysis, feature engineering, training and validation splits, model selection, training, overfitting, underfitting, evaluation metrics, deployment, inference, monitoring, drift, retraining, versioning, privacy, and reproducibility. Use one consistent practical example throughout the explanation and clearly distinguish training from inference.

Prompt 5 — Incident Response

Prepare a detailed incident-response guide for a software team operating a high-traffic web service. Explain detection, alert triage, severity classification, incident command, communication, diagnosis, mitigation, rollback, feature flags, logs, metrics, distributed traces, dashboards, database investigation, external dependency failures, customer communication, recovery validation, post-incident review, and preventive actions. Include a realistic example where an API deployment causes elevated error rates and database load. Keep the guide practical and operational.

Prompt 6 — Containerization and Application Deployment

Write a comprehensive explanation of containerization and modern application deployment. Begin with the problems containers solve and explain images, containers, registries, Dockerfiles, layers, volumes, networking, environment variables, secrets, resource limits, health checks, and logging. Then explain how container orchestration systems schedule and manage applications, including replicas, rolling deployments, service discovery, autoscaling, failure recovery, and persistent storage. Finish by walking through a realistic deployment of a frontend, API, and database.

Profile 4 — Instruct: Reasoning

Prompt 1 — Factory Production Optimization

A company operates three factories A, B, and C. Factory A can produce 1,200 units per day at \$18 per unit, B can produce 900 units per day at \$21 per unit, and C can produce

1,500 units per day at \$16 per unit. Daily demand is 3,000 units. However, at least 600 units must come from B because of a contract, C can operate at no more than 80 percent capacity because of maintenance, and using more than 1,000 units from A adds an overtime cost of \$5 for every A unit above 1,000. Determine a lowest-cost feasible production allocation. Show the calculations, verify every constraint, calculate total cost, and explain why shifting production among factories would or would not reduce the cost.

Prompt 2 — Project Scheduling and Critical Path

A company must schedule six projects A through F. Their durations in days are A=4, B=7, C=3, D=6, E=5, and F=2. B and C can begin only after A finishes. D requires B. E requires both B and C. F requires both D and E. Assume unlimited workers, but no project can begin before all of its dependencies are complete. Determine the earliest start and finish time for every project, identify the critical path, calculate the minimum total completion time, and then determine what changes if B takes three additional days. Show the reasoning step by step.

Prompt 3 — Retail Pricing and Profit

An online retailer sells a product for \$480. The retailer first applies a 15 percent promotional discount. A customer then uses a coupon worth 10 percent of the already-discounted price. Shipping costs \$24 unless the price after both discounts is at least \$375, in which case shipping is free. Finally, 13 percent sales tax is applied to the merchandise price after discounts plus any shipping charge. The retailer's original cost for the product is \$260 and it pays the shipping charge itself when shipping is free. Calculate the customer's final payment, the retailer's gross profit before other expenses, and the effective percentage discount relative to the original \$480 sticker price. Show every calculation and explain the order of operations.

Prompt 4 — Data Center Energy Efficiency

A data center has a continuous IT load of 600 kW. Its initial power usage effectiveness, or PUE, is 1.50. An efficiency project reduces PUE to 1.25 but requires a \$900,000 upfront investment. Electricity costs \$0.12 per kWh during the first year and rises by 4 percent at the beginning of each subsequent year. Assume 8,760 operating hours per year and that the IT load remains constant. Calculate annual facility energy consumption before and after the upgrade, first-year energy and cost savings, and the cumulative nominal electricity-cost savings over five years. Determine whether the \$900,000 investment has paid back within five years and estimate when simple payback occurs. Show all steps and assumptions.

Prompt 5 — Delivery Fleet Optimization

A delivery company has three trucks. Truck A can carry 8 tonnes and costs \$140 per trip, Truck B carries 12 tonnes and costs \$190 per trip, and Truck C carries 20 tonnes and costs \$310 per trip. The company must move exactly 120 tonnes. At least two trips must use Truck A, no more than four trips may use Truck C, and the total number of trips may not exceed ten. Determine a minimum-cost feasible combination of integer trips for A, B, and C. Systematically evaluate the constraints, show the calculations, report total capacity and total cost, and explain why no cheaper feasible integer combination exists.

Prompt 6 — Five-Year Server TCO

A company is comparing two server systems for a five-year deployment. System A costs \$18,000 initially, consumes an average of 420 watts, and requires \$1,200 annual maintenance. System B costs \$26,000 initially, consumes 250 watts, and requires \$700 annual maintenance. Both run continuously 8,760 hours per year. Electricity costs \$0.15 per kWh in year one and increases by 5 percent at the start of each following year. At the end of year five, System A has a resale value of \$2,000 and System B has a resale value of \$4,000. Calculate year-by-year electricity costs, maintenance costs, total five-year nominal ownership cost after resale value, the difference between the systems, and the approximate electricity price at which their five-year costs would become equal if all other assumptions stayed unchanged. Show the reasoning clearly.