

Software Engineer Tech Lead

Role Summary

We are seeking a hands-on, full-stack Tech Lead to work across our entire engineering platform. The Tech Lead will design, implement, review, test, release, and support software across frontend, backend, infrastructure, and integrations.

In addition to contributing broadly, each Tech Lead serves as the primary technical owner for selected areas of the engineering landscape. Area ownership means leading architectural direction, significant design decisions, technical quality, documentation, risk management, and long-term evolution. It does not mean working exclusively in that area or being its sole contributor.

The ideal candidate is a strong engineering generalist. Deep experience with modern frontend development (particularly React, TypeScript, frontend architecture, accessibility, testing, and performance) is a significant advantage.

The Tech Lead owns delivery **end-to-end across the stack**; from frontend to backend to supporting services.

Responsibilities

Full-Stack Technical Leadership

- Lead design and implementation across the full stack (frontend, backend, APIs, and supporting services)
- Design and maintain clear boundaries between frontend and backend systems, ensuring well-defined interfaces and minimal coupling between layers
- Make architectural decisions that optimize for simplicity, performance, and maintainability
- Perform code reviews across the stack with consistent standards

Polyglot Engineering & System Design

- Select appropriate languages, frameworks, and tools based on the problem space
- Guide the team in understanding tradeoffs between different technologies
- Introduce new technologies when they provide clear advantages in performance, safety, or developer efficiency
- Avoid unnecessary complexity by matching solutions to business needs

System Boundaries & Interface Contracts

- Define and enforce clear contracts between frontend and backend systems (e.g., APIs, schemas), ensuring each can evolve independently without breaking downstream consumers
- Establish API standards using OpenAPI (or equivalent) as the source of truth for backend interfaces
- Ensure API contracts are explicitly versioned when introducing breaking changes
- Promote the use of typed clients or shared schemas to reduce integration errors
- Require that any contract changes are documented, reviewed, and backward-compatible where feasible (or versioned when not)
- Prevent tight coupling by avoiding implicit dependencies (e.g., undocumented fields, shared assumptions, or UI-driven backend behavior)

Technical Documentation & Design Ownership

- Produce high-quality design documents that demonstrate clear reasoning, well-defined system boundaries, explicit contracts, and thoughtful tradeoffs
- Own the technical soundness of proposed solutions, not just their documentation
- Ensure designs are:
 - Correct and feasible
 - Aligned with system constraints and long-term architectural direction
 - Structured for incremental, iterative delivery
- Translate high-level architectural direction into concrete, buildable designs for the team
- Maintain supporting documentation including playbooks, runbooks, and system-level documentation
- Ensure critical workflows and systems are documented and not dependent on a single individual
- Establish documentation as a required part of delivering work, not an afterthought

Iterative Development & Delivery

- Drive incremental, end-to-end delivery of features (UI → API → data)
- Break down work into small, testable, and deployable units
- Promote rapid feedback loops through:
 - Trunk-based development
 - Feature flags
 - Continuous integration
- Prevent large, high-risk releases by enforcing iterative delivery practices

AI-Accelerated Engineering

- Treat AI as a core part of the development workflow, using it to accelerate delivery while maintaining high standards of correctness, security, and maintainability

- Leverage AI tools to improve development speed, code quality, and overall engineering outcomes
- Use AI to assist with:
 - Code generation and refactoring
 - Test creation and coverage improvement
 - Documentation (design docs, playbooks, summaries)
 - Debugging and root cause analysis
- Apply strong judgment when using AI:
 - Validate correctness and security of generated outputs
 - Avoid introducing unnecessary complexity or hidden dependencies
- Continuously evaluate and introduce AI-assisted workflows that improve team productivity
- Share effective AI usage patterns with the team and establish best practices

Delivery & Execution

- Own day-to-day execution using Kanban workflows
- Ensure work is clearly defined and flows efficiently
- Monitor throughput and cycle time to identify bottlenecks
- Collaborate directly with stakeholders to clarify requirements and priorities

Operational Excellence

- Ensure robust testing across the stack (unit, integration, end-to-end)
- Maintain system performance, reliability, and security
- Drive improvements in development workflows, tooling, and deployment processes

Qualifications

- 8+ years of software engineering experience
- Strong full-stack experience (backend + modern frontend frameworks)
- Deep expertise in Python and API development (FastAPI preferred)
- Experience working across multiple languages or tech stacks (e.g., Python, TypeScript, Go)
- Proven experience leading engineers or acting as a technical lead
- Strong system design and architectural decision-making skills
- Experience writing clear, high-quality technical documentation (design docs, diagrams, playbooks)
- Experience with trunk-based development and iterative delivery practices
- Experience effectively using AI tools to improve development speed and quality

What Success Looks Like

- Features are delivered end-to-end in small, reliable increments
- Frontend and backend systems evolve independently through well-defined contracts
- API contracts are clear, documented, and rarely cause regressions
- Design documents are clear, technically sound, and enable engineers to execute without ambiguity
- Breaking changes are intentional, versioned, and well-communicated
- The team uses the right tools for each problem, not just familiar ones
- Knowledge is distributed across the team with no single points of failure
- Engineers are productive, unblocked, and continuously improving
- AI usage meaningfully improves development speed, quality, and documentation

How to Apply

Send your resume plus any other supporting materials to careers@smylfitness.com and put “**Application: ENG-260601 - Software Engineer Tech Lead**” in the subject line.