

Fictional Sample - Personalized Report

Software Developer - your actual work

49 / 100

VERY HIGH EXPOSURE

More exposed than 87% of analyzed occupations. Occupation baseline: 41/100.

Confidence: MEDIUM

Reduced because the description was short or unspecific.

0 LOWER TASK EXPOSURE

HIGHER TASK EXPOSURE 100

This score estimates how exposed the tasks in a role are to current and near-term AI capabilities. It does not predict whether a specific person will lose a job.

Executive summary

Your AI task exposure score is 49, placing you in the Very High band-notably above the Software Developer baseline of 41. This elevation reflects that your work is concentrated in two augmentable tasks that together account for 85% of your time. Your seniority and the specific mix of coding and schema work you do make you more exposed to AI adoption than the typical developer in your role.

Your personalized score

Your score is driven primarily by 'Write and review application code' (70% of your work, exposure 55) and 'Design database schemas and migrations' (15% of your work, exposure 45). Both are classified as augmentable-AI tools can meaningfully assist but not fully replace the human judgment involved. The remaining 15% spent on 'Monitor system functioning and incident response' (exposure 25) is more durable and acts as a stabilizing factor. Because your largest task is code-focused and AI code generation is actively mature, your overall exposure sits higher than the occupation average.

You score 8 points above the baseline (49 vs. 41) and sit at the 87th percentile rather than the 67th. The gap exists because your task distribution-heavy on application code-aligns closely with where large language models and code assistants have gained the most traction. A typical software developer might have more incident response or architectural oversight work; you have less of that buffer.

Your task matrix

TASK	SHARE	CLASSIFICATION	EXPOSURE
Write and review application code	70%	Augmentable	55
Design database schemas and migrations	15%	Augmentable	45
Monitor system functioning and incident response	15%	Durable	25

Unmatched tasks are excluded from the score and reflected in confidence instead. Exposure values are fixed, versioned rubric scores (see final page).

Next 7 days

- **Document your incident response and system monitoring work (effort: low, impact: medium)**
Your 'Monitor system functioning and incident response' work has the lowest AI task exposure score (25) among your tasks. Making this expertise visible and concrete-through incident summaries, postmortems you've led, or monitoring improvements you've implemented-positions your most durable contributions prominently.
- **Audit your code review process and standards (effort: low, impact: medium)**
'Write and review application code' is your largest task (70% of your work) and has an exposure score of 55. Review whether your code review practice emphasizes judgment, architectural insight, and mentorship-areas less substitutable by automation-rather than mechanical checks.
- **Identify one schema or migration decision you led (effort: low, impact: low)**
'Design database schemas and migrations' (15% of work, exposure 45) benefits from clear examples of trade-off reasoning, scalability choices, or complexity you resolved. One concrete case study prepares you for both immediate conversations and longer-term pivots.

Your human moat

- **Monitor system functioning and incident response**
Diagnosing production failures and making real-time decisions under uncertainty require pattern recognition, stakeholder communication, and accountability that remain difficult to fully automate. This work anchors your role.
- **Design database schemas and migrations**
While AI can suggest schema structures, the reasoning-data consistency trade-offs, performance under your specific workload, migration safety-still demands human judgment grounded in your system's constraints and history.

Adopt AI before it is imposed

- **Code generation in 'Write and review application code'**
LLMs like GitHub Copilot and ChatGPT can now draft functions, tests, and refactorings at speed. Instead of resisting this, expect to spend more time reviewing, testing, and integrating AI suggestions than writing from scratch. Your review rigor becomes the differentiator.
- **Schema design assistance and documentation**
AI tools can generate migration scripts, normalize schema proposals, and document relationships. Use them to explore alternatives faster, but retain final sign-off on correctness and performance implications-especially for production systems.
- **Incident diagnostics and runbook automation**
AI can help correlate logs, suggest probable causes, and draft runbooks. Lean into this for routine diagnostics, freeing you to focus on novel failures and post-incident learning-the parts that still require your judgment.

Your 30-day plan

- **Map your architectural and mentoring contributions (effort: medium, impact: high)**
At senior level, your leverage lies in decisions and guidance that span multiple projects. Identify 3-4 examples where you shaped direction, reduced technical debt, or grew teammates-these are harder to augment and valuable in any senior role.
- **Strengthen your incident-response leadership narrative (effort: medium, impact: medium)**
Incident response and system monitoring (your most durable work) often scales to leadership roles. Document the scope, criticality, and team coordination in 2-3 recent incidents to show readiness for broader responsibility.
- **Explore Engineering Manager transition informally (effort: medium, impact: medium)**
The adjacent role available to you is Engineering Manager (skill overlap 82%, though with a different exposure profile). Have conversations with managers in your network about what the role entails, what gaps you'd need to close, and whether mentorship is already a strength of yours.

■ Assess your hands-on coding depth vs. breadth preference (effort: low, impact: high)

Your personal task exposure score (49) is higher than the occupation baseline (41), partly because 'Write and review application code' dominates your day. Clarify whether you want to stay deep in hands-on coding (and invest in architectural and review expertise) or transition toward people/system leadership.

■ 90-day direction

■ Define your senior-level differentiation

By end of Q2/Q3, articulate whether your next 2-3 years focus on deepening as a staff or principal engineer (leveraging architectural, code-review, and mentoring strengths) or pivoting toward Engineering Manager. Your durable work in incident response and system oversight fits both paths.

■ Build or refresh your external profile

Update your resume, LinkedIn, or portfolio to emphasize the architectural and leadership dimensions of your three core tasks. Ensure incident response, schema decisions, and code review are framed as senior-level judgment, not just execution.

■ Lower-exposure adjacent roles

■ Engineering Manager - 30/100 - 19 points lower, ~82% skill overlap

Skill overlap is strong (82%), and your incident-response and system-monitoring work translates directly to on-call leadership and team resilience. The exposure profile differs-management has lower AI task exposure in some areas but requires people skills and decision-making you're already exercising. Key gap: do you have or want mentoring, conflict resolution, and hiring experience? This pivot is plausible if you enjoy developing people.

■ Resume positioning

- Lead or co-own incident response and postmortem processes that improved system reliability or mean-time-to-recovery. Quantify scope (e.g., systems impacted, team size involved, severity).
- Emphasize architectural and design contributions in schema and migration work-decisions made, trade-offs evaluated, scaling challenges solved-rather than just execution.
- Frame code review as a force multiplier: mentorship provided, quality standards set, security or performance patterns established across the team.
- Highlight cross-project or cross-team systems thinking, long-term technical direction you've shaped, or technical debt reduction you've championed.

■ Skill priorities

■ System design and scalability reasoning

Complements your database and architecture work. Deepens your competitive advantage in 'Design database schemas and migrations' and makes you less dependent on code-writing velocity.

■ Mentoring and knowledge transfer

Leverages your senior level and your code-review responsibilities. Opens pathways to staff/principal roles or management. Less automatable than individual coding output.

■ Observability, monitoring, and debugging practices

Sharpens your 'Monitor system functioning and incident response' work-your most durable task. Expertise in designing monitoring and root-cause workflows is valuable at senior level and harder to augment.

■ People leadership or team coordination (if management interests you)

The Engineering Manager pivot requires hands-on practice: running meetings, delegating, handling disagreement, and developing others. Start with small leadership opportunities now.

Methodology & versions

Your personalized score is a deterministic aggregate of fixed per-task exposure values: $\text{score} = \frac{\text{sum}(\text{task share} \times \text{task exposure})}{\text{sum}(\text{shares})}$. The task exposure seed was model-scored under a versioned, anchored rubric and published as fixed data; the report pipeline did not invent or alter those values. Where available, AI extracted described tasks and drafted explanations; deterministic fallbacks are labeled above. See jobairisk.com/methodology. Versions: score jr-v1, data 2026.07.12-r2, prompt p1, narrative model claude-haiku-4-5.