Frontend Developer Interview Scorecard



Candidate Name: Role Interviewed: Interviewer: Date:	
Dimensions	
• Reliability & Incident	t Management — Score (1–5):
1-2: Fails to triage in Follows runbooks, connotes. 4: Leads resp	cidents, delays response, or ignores runbooks; causes repeated outages. 3: ontains incidents, and performs timely mitigation with documented post-incident onse across teams, reduces MTTR, and drives effective postmortems with clear nes incident strategy, enforces SLOs/error budgets, and eliminates classes of
System Architecture	e & Scalability — Score (1–5):
1-2: Designs brittle s3: Designs redundarsystems for predictal	ringle-point solutions and lacks capacity planning or failure domain awareness. In the components with capacity estimates and basic failure isolation. 4: Architects ble scale, identifies failure modes, and proposes resilient patterns. 5: Owns ecture decisions, influences platform roadmaps, and drives large-scale scalability
 Automation & Infras 	tructure as Code — Score (1–5):
1-2: Performs manual infrastructure. 3: Impleasic testing. 4: Auto	al changes frequently and lacks idempotent automation or versioned elements IaC for services and environments with repeatable deployments and emates runbooks, CI/CD, and rollback procedures; enforces policy as code. 5: mation strategy, creates resilient self-healing workflows, and reduces
Observability & Mon	itoring — Score (1–5):
1-2: Lacks meaningf dashboards, sets ale alerts, reduces alert	ul metrics, noisy alerts, and insufficient logs to diagnose issues. 3: Creates erts, and collects logs/traces sufficient for troubleshooting. 4: Defines SLO-based fatigue, and instruments end-to-end traces for latency and errors. 5: Implements ty, drives SLO adoption across teams, and ties telemetry to business outcomes.

zythr.com Page 1 of 1

ZYTHR

1-2: Writes untested, hard-to-read scripts; struggles to debug production problems. 3: Produces readable, tested code and uses debugging tools to identify root causes. 4: Optimizes performance hotspots, performs code reviews that improve reliability, and writes reusable libraries. 5: Drives engineering disciplines that prevent classes of bugs and mentors teams on robust coding practices.

Collaboration & Communication — Score (1–5):

1-2: Communicates unclearly in incidents and fails to align stakeholders or document decisions. 3: Communicates status during incidents, writes clear runbooks, and aligns with downstream teams. 4: Facilitates cross-team technical discussions and negotiates trade-offs effectively. 5: Influences product and engineering priorities through clear, data-driven communication and consensus building.

Mentorship & Knowledge Sharing — Score (1–5):

1-2: Does not share knowledge, hoards runbooks, or avoids mentoring opportunities. 3: Provides constructive code reviews, updates documentation, and mentors junior engineers occasionally. 4: Regularly coaches peers, leads learning sessions, and improves team on-call capabilities. 5: Builds scalable training, creates onboarding programs, and measurably raises team reliability competence.

Overall Evaluation

Strengths Observed:

Concerns / Weaknesses:

Recommendation (Yes / No / With Reservations):

Final Score (Avg / Weighted):

zythr.com Page 1 of 2