**Frontend Developer Interview Scorecard**

Candidate Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Role Interviewed For: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Interviewer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| **Dimension** | **Guidance** | **Score (1–5)** |
| Reliability & Incident Management | 1-2: Fails to triage incidents, delays response, or ignores runbooks; causes repeated outages.
3: Follows runbooks, contains incidents, and performs timely mitigation with documented post-incident notes.
4: Leads response across teams, reduces MTTR, and drives effective postmortems with clear action items.
5: Defines incident strategy, enforces SLOs/error budgets, and eliminates classes of incidents through systemic change. |  |
| System Architecture & Scalability | 1-2: Designs brittle single-point solutions and lacks capacity planning or failure domain awareness.
3: Designs redundant components with capacity estimates and basic failure isolation.
4: Architects systems for predictable scale, identifies failure modes, and proposes resilient patterns.
5: Owns cross-service architecture decisions, influences platform roadmaps, and drives large-scale scalability initiatives. |  |
| Automation & Infrastructure as Code | 1-2: Performs manual changes frequently and lacks idempotent automation or versioned infrastructure.
3: Implements IaC for services and environments with repeatable deployments and basic testing.
4: Automates runbooks, CI/CD, and rollback procedures; enforces policy as code.
5: Drives platform automation strategy, creates resilient self-healing workflows, and reduces operational toil significantly. |  |
| Observability & Monitoring | 1-2: Lacks meaningful metrics, noisy alerts, and insufficient logs to diagnose issues.
3: Creates dashboards, sets alerts, and collects logs/traces sufficient for troubleshooting.
4: Defines SLO-based alerts, reduces alert fatigue, and instruments end-to-end traces for latency and errors.
5: Implements proactive observability, drives SLO adoption across teams, and ties telemetry to business outcomes. |  |
| Software Engineering & Debugging | 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 | 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 | 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):**