**Frontend Developer Interview Scorecard**

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

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

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

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

|  |  |  |
| --- | --- | --- |
| **Dimension** | **Guidance** | **Score (1–5)** |
| Automation Scripting | 1-2: Writes brittle, repetitive scripts and struggles with basic syntax or constructs.
3: Implements reliable scripts for simple flows and uses basic reusability.
4: Designs modular, parameterized scripts that reduce duplication.
5: Creates reusable libraries and abstractions that speed future test development. |  |
| Testing Fundamentals | 1-2: Cannot explain basic testing concepts or write clear test cases.
3: Understands test types and writes clear positive and negative cases.
4: Identifies edge cases and maps tests to explicit requirements.
5: Defines simple coverage checks and closes clear gaps in coverage. |  |
| Test Design & Analysis | 1-2: Creates ad-hoc tests without structuring or prioritizing cases.
3: Designs prioritized suites and selects representative test data.
4: Builds data-driven suites and optimizes case selection by risk.
5: Defines test strategy for features and influences release decisions. |  |
| Framework & Tooling | 1-2: Depends on manual UI steps and cannot configure the framework.
3: Uses existing framework features and adds basic helper utilities.
4: Extends the framework and integrates useful libraries for reliability.
5: Proposes and implements framework improvements adopted by the team. |  |
| Debugging & Troubleshooting | 1-2: Cannot isolate failures or reproduce issues consistently.
3: Reproduces failures and provides useful logs and reproduction steps.
4: Root-causes intermittent issues and suggests reliable fixes.
5: Creates tooling or processes that reduce flakiness across suites. |  |
| CI/CD & Test Ops | 1-2: Unfamiliar with CI or cannot run tests in pipeline environments.
3: Integrates tests into CI and understands pipeline failure signals.
4: Optimizes pipelines for faster execution and more reliable runs.
5: Automates test reporting and exposes health metrics to the team. |  |
| Communication & Collaboration | 1-2: Provides unclear bug reports and struggles to accept feedback.
3: Writes clear bug reports and updates stakeholders on progress.
4: Proactively coordinates with developers and QA to unblock work.
5: Leads triage or cross-team discussions that improve delivery. |  |

 **Overall Evaluation**

**Strengths Observed:**

**Concerns / Weaknesses:**

**Recommendation (Yes / No / With Reservations):**

**Final Score (Avg / Weighted):**