**Frontend Developer Interview Scorecard**

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

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

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

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

|  |  |  |
| --- | --- | --- |
| **Dimension** | **Guidance** | **Score (1–5)** |
| Technical Knowledge | 1-2: Struggles to write or run basic backend code; frequent syntax and runtime errors.
3: Writes correct code for small tasks with occasional guidance; understands core language features.
4: Independently implements features using language idioms and standard libraries with few errors.
5: Applies best practices and performance-aware patterns; helps others choose appropriate language features. |  |
| Code Quality & Testing | 1-2: Commits code lacking tests and with inconsistent style; ignores linting and review feedback.
3: Produces readable code and basic unit tests; follows repository style and addresses review comments.
4: Writes well-structured code with good test coverage and meaningful test cases; CI passes consistently.
5: Designs testable modules, covers edge cases, and improves testing practices or CI reliability. |  |
| Debugging & Troubleshooting | 1-2: Cannot reproduce bugs or relies on others to diagnose basic failures.
3: Reproduces issues and uses logs or stack traces to identify causes with guidance.
4: Quickly isolates root causes, proposes fixes, and verifies resolution in tests or staging.
5: Anticipates failure modes, adds diagnostics or alerts, and prevents recurrence. |  |
| API Design & Integration | 1-2: Creates inconsistent or breaking endpoints and ignores request/response contracts.
3: Implements clear APIs for simple endpoints and follows existing contracts and error conventions.
4: Designs stable, versioned APIs that handle errors and edge cases; documents usage.
5: Shapes API guidelines, improves backward compatibility, and provides integration examples. |  |
| Collaboration & Communication | 1-2: Rarely asks for clarification, writes unclear PRs, and misses team norms or deadlines.
3: Communicates status, writes clear PR descriptions, and asks needed questions in a timely way.
4: Proactively coordinates with teammates, responds to reviews constructively, and documents decisions.
5: Leads small discussions, clarifies trade-offs, and helps align teammates on implementation plans. |  |
| System Design & Architecture | 1-2: Cannot explain high-level component interactions or trade-offs for a feature.
3: Explains simple service boundaries and data flow for small features.
4: Chooses appropriate patterns for scalability and reliability with some guidance.
5: Contributes useful suggestions to architecture discussions and proposes improved designs. |  |
| Learning & Ownership | 1-2: Avoids unfamiliar tasks and requires constant direction to make progress.
3: Learns from feedback and completes assigned tasks with occasional help.
4: Takes ownership of features, seeks feedback, and acquires new skills quickly.
5: Drives improvements, proactively learns new technologies, and helps onboard others. |  |

 **Overall Evaluation**

**Strengths Observed:**

**Concerns / Weaknesses:**

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

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