INTERVIEW EXPERIENCE BROWSERSTACK

My Interview Experience for an SDET Role :

Round 1: Resume Shortlisting

- Profile matched role requirements in testing, automation, CI/CD, and backend exposure.

Round 2: Online Assessment (DSA)

- Duration: 60 minutes.
- Format: 3 coding problems (2 easy–medium, 1 hard).

Round 3: Machine-Coding Exercise

- Duration and format: 3 hours, build a small backend service with a few REST endpoints; emphasis on working software over completeness.
- You were allowed to use google but using AI tools was strictly prohibited also you have to share your screen the entire time and some or the other one is continuously watching.
- Problem shape: Control and inspect an external process/resource via HTTP; include start/stop/status/cleanup flows; keep the API predictable and safe.
- Design expectations:
    - API contracts: Well-named endpoints, clear request/response shapes, consistent HTTP status codes (200/201/400/404/409/500) and error bodies.
    - Lifecycle management: Idempotent operations (repeated start/stop shouldn't break state), graceful handling of already-running/already-stopped cases, and timeouts.
    - State and persistence: Minimal in-memory registry plus lightweight persistence or recovery notes so restarts don't lose track; avoid hidden global state.
    - Reliability and safety: Input validation, defensive checks before executing actions, and cleanup paths to avoid orphaned resources.
    - Observability: Structured logs around key transitions (create/stop/fail), simple health/readiness checks if time permits.
    - Code organization: Separation of concerns (routes/controllers/services), small functions, and unit-testable core logic.
- What's typically assessed: correctness of flows, clarity of design, robustness to edge cases, readability, and how trade-offs are communicated.

Round 4: Engineering Manager Discussion

- Technical depth:
    - Project walkthroughs: scope, stack, architecture decisions, and measurable outcomes.
    - Automation approach: test pyramid design (unit → integration/contract → E2E), data seeding/fixtures, strategies to reduce flakiness (selectors, waits, retries budgeted by policy).
    - CI/CD practices: branching strategy, mandatory checks (lint, tests, coverage thresholds), strong knowledge of github (how to work in a dev team).

- Problem solving and ownership:
    - Handling production issues: triage, root-cause analysis, containment, and prevention (runbooks, alerts, guardrails).
    - Trade-offs: performance vs. reliability, coverage vs. build time, speed vs. review quality.
- Collaboration:
    - Working with devs, QA, and product; writing clear tickets/specs; communicating risk and timelines.
- Evaluation lens: depth over tool names, ability to justify decisions, clarity and structure in explanations.

Round 5: Director Conversation

- Scope and impact:
    - High-level challenges: examples where constraints were unclear, requirements shifted, or timelines were tight; how decisions were made.
    - Definition of quality: reliability, user impact, and how to measure it (defect escape rate, flakiness rate, lead time, MTTR).
- Alignment and motivation:
    - Why this company/team: connection to product, users, and engineering culture.
    - Role trajectory: how SDET work amplifies delivery speed and reduces risk; areas of ownership envisioned.
- Forward-looking topics:
    - Testing with AI in the loop: where AI helps (authoring, triage, flaky detection) and where human judgment remains essential (test design, risk analysis).
    - Building a quality culture: lightweight processes, actionable dashboards, and clear “done” definitions.
- Evaluation lens: strategic thinking, communication crispness, and cultural fit.

HR Touchpoints

- Purpose: scheduling, clarity on next-round focus, logistics (tools, platform), and timelines.
- Expectation setting: brief summaries of what each round emphasizes and when to expect outcomes.