# QA Strategy for SaaS Platform

As the sole QA engineer in a SaaS startup, I would take a structured approach to ensure quality across the entire platform, integrating testing with the SDLC and maximizing coverage quickly.

## 1. QA Approach for the Entire Platform

* **SDLC Integration:** Embed QA activities from day one, including requirements review, early test case design, and participation in sprint planning to catch issues early.
* **API Testing:** Use REST-assured or Postman to test endpoints, validate request/response schemas, and perform automated regression on API functionality.
* **Web Frontend Testing:** Apply cross-browser and responsive testing using Playwright or Selenium. Include both functional and UI/UX checks.
* **Hybrid Mobile App Testing:** Utilize Appium to automate core workflows, complemented by manual exploratory testing for gestures and native interactions.
* **USSD App Testing:** Create a combination of automated USSD flow simulations (via terminal scripts) and manual testing to cover real network scenarios.

## 2. Plan to Achieve 100% Test Coverage in the First 30 Days

* **Day 1–7:** Map out all modules, APIs, and user flows. Identify critical paths and high-risk areas.
* **Day 8–15:** Write automated test scripts for core APIs and critical frontend workflows. Start smoke tests for mobile and USSD.
* **Day 16–25:** Expand coverage to edge cases, error handling, and less-used flows. Integrate tests into CI/CD pipelines.
* **Day 26–30:** Review gaps, add regression suites, and ensure test execution is automated and repeatable across all platforms.

## 3. Tools and Processes to Introduce

* **Version Control & CI/CD:** GitHub/Bitbucket with pipelines to trigger automated tests on each commit.
* **Test Automation:** Playwright for web, Appium for mobile, Postman/REST-assured for APIs.
* **Test Management:** Jira/TestRail to track test cases, results, and defects.
* **Monitoring & Reporting:** Generate automated reports and dashboards for quick visibility into quality metrics.
* **Code Review Integration:** Collaborate with developers to enforce testability and quality gates.

## 4. Ensuring Quality Across All Products

* Maintain **consistent testing standards** across web, mobile, API, and USSD apps.
* Conduct **regular exploratory sessions** to catch unknown issues.
* Establish **daily/weekly automated test runs** in CI/CD to catch regressions early.
* Share **quality metrics with the team** to drive accountability and continuous improvement.
* Implement **risk-based testing** to prioritize high-impact areas while maintaining full coverage.

By combining structured automation, risk assessment, and integration with the development process, quality can be maintained across all products while scaling test coverage rapidly within the first month.