Phase 2: Progressive Integration Testing (API-level)

As the system evolves and core functionality stabilizes, API-level integration tests will be introduced as a fast, reliable way to validate backend behavior, reduce dependency on manual UI testing, and lay the foundation for scalable automation. Integration testing will also serve as the initial smoke testing layer, executed before any UI-based testing begins.

If integration tests pass, the QA workflow continues to UI-level manual or automated testing.

If integration tests fail, QA is paused, and the issue is escalated to engineering without wasting time on full frontend validation.

Coverage Strategy  
The integration test suite will focus on:

• Verifying all primary API calls in the proposal generation pipeline (e.g., uploads, generate, and retrieve)

• Validating backend behavior in isolation from the UI

• Expanding coverage to include corner cases and edge input conditions over time, to relieve pressure from the manual testing process

Tooling

Integration testing will be implemented using the following stack:

• Postman – initial collections and quick exploratory validation

• Python – main language

• Pytest + Requests + JSON Schema – core framework for API integration tests

• Allure – test reporting ??

**Phase 3: Selective Test Automation**

Automation will be introduced gradually and selectively, building on the foundation of manual UI testing (Phase 1) and automated integration testing (Phase 2).

The goals of this phase are to:

* Replace repetitive manual test cases with stable automated scenarios
* Strengthen CI/CD reliability through automated smoke and regression checks

**Tooling**

* Python + Pytest
* Playwright
* **Allure – test reporting ??**