# Automation & API Testing Strategy – T-Call Platform (Goghr)

## 1. Automation Test Plan

The automation test plan is designed to reduce manual testing effort, increase coverage, and ensure repeatable validation of T-Call’s core features including dashboards, agent creation, and call workflows.

Framework & Tools:

* ✅ UI Automation: Selenium with Python (or Playwright for advanced UI control)
* ✅ API Automation: Pytest with requests or Postman/Newman for CI integration
* ✅ CI/CD Integration: GitHub Actions or Jenkins for scheduled/triggered test runs
* ✅ Reporting: Allure Reports / HTML Reports / JUnit XML format for dashboards

Suggested Automation Scope:

* ✅ Login/Logout Flow for Admin and Clients
* ✅ Agent Creation Workflow (Retail & Eleven Labs)
* ✅ Call Session Initialization and Completion
* ✅ Dashboard KPIs and Stats Display
* ✅ Notification Triggers (Subscription updates, Errors)
* ✅ WhatsApp Template CRUD and Messaging
* ✅ Regression Suite for stable endpoints and UI flows

## 2. API Testing Strategy

T-Call interacts with multiple third-party APIs. Validating API reliability, response structure, and failure handling is crucial.

Key APIs to Test:

* ✅ Retail API – Agent creation, session handling
* ✅ Eleven Labs API – Voice session management, WebSocket triggers
* ✅ Twilio API – Call trigger, SMS/voice notifications, webhook responses

API Testing Checklist:

* ✅ Verify status codes (200, 400, 401, 403, 500 etc.)
* ✅ Validate request/response schemas and data types
* ✅ Auth Token handling – expiration and refresh
* ✅ Timeout, Retry, and Error Handling for failed requests
* ✅ Webhook Responses and Signature Validation (for Twilio)
* ✅ Load testing for high-volume API endpoints (e.g., batch voice sessions)

Recommended Tools:

* ✅ Postman (manual testing, environment configs)
* ✅ Newman (CI/CD integration for Postman collections)
* ✅ Pytest + Requests (automated API tests)
* ✅ Locust or JMeter (API performance and load testing)