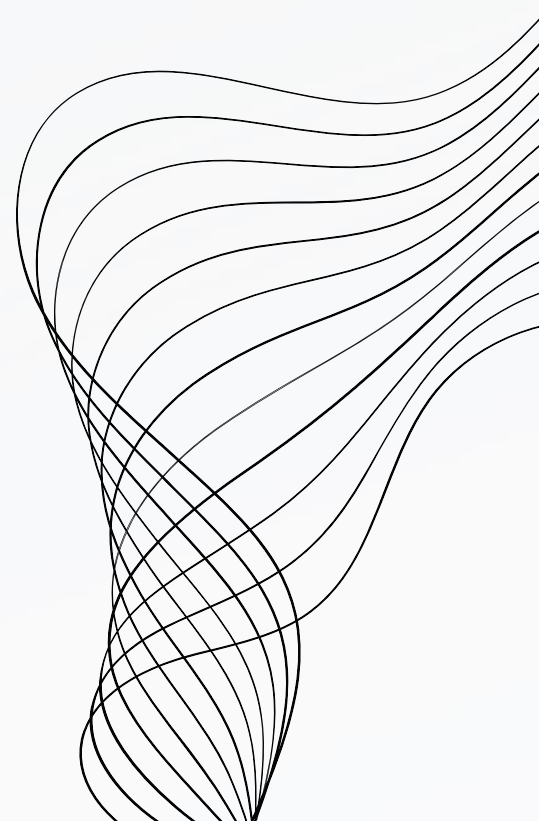




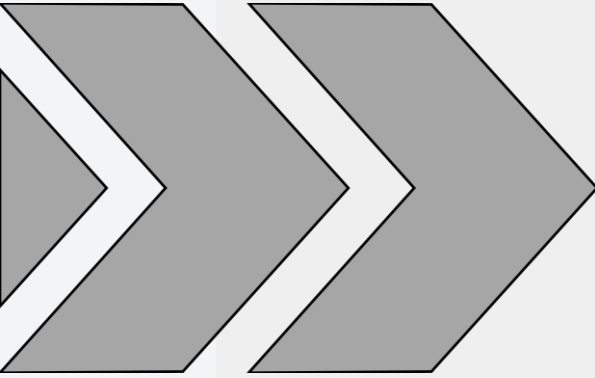
IQE GENERATIVE AI HACKATHON 2024

BYTE BUILDERS

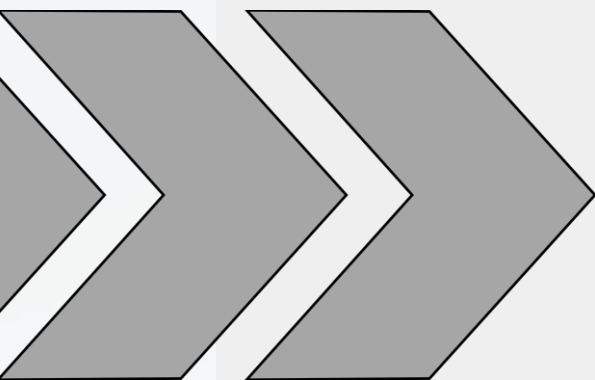
Name	Email
Manali Shantanu Lawate	manalishantanu.l@infosys.com
Rajat Dandoti	rajat.dandoti@infosys.com
Seshan Nagaraj	seshan.nagaraj@infosys.com



PROBLEM STATEMENT



Frequent locator changes due to UI adjustments can cause **25–50%** of test scripts to fail, resulting in a **~40% increase in maintenance re-work** and **consuming valuable resources**.



Consequently, this necessitates manual execution of failed test scripts during the regression testing phase, **impacts efficiency & effectiveness** of the testing process by **30–60%**.

Background:

- The client operates web applications and uses a **Selenium-based test automation framework** for efficient Quality Engineering.
- This automation suite is executed across various environments during key testing phases, including system testing and end-to-end testing.
- In an Agile model, frequent **feature enhancements lead to UI changes**, impacting the stability of automated tests.

GOALS AND OBJECTIVES

Our use case focuses on a **GenAI-led locator self-healing solution** for UI automation frameworks. This solution automatically repairs failed locators occurred due to frequent UI changes, aiming to reduce script failures and enhance testing efficiency. By generating concise reports, it empowers the Quality Engineering team to maintain high-quality software delivery in an Agile environment.

Replace Locator

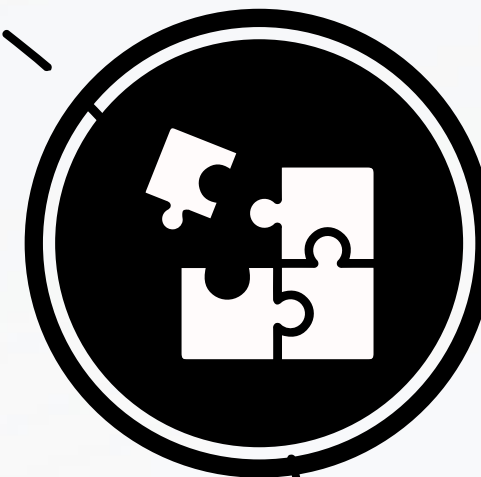
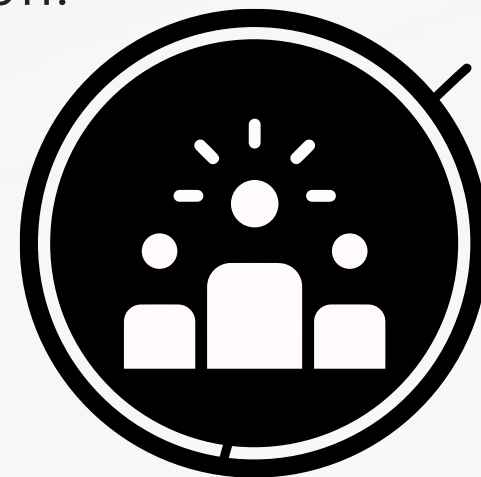
Seamlessly update the failed locators with the corrected locators to ensure continued test execution.

Locator Healing

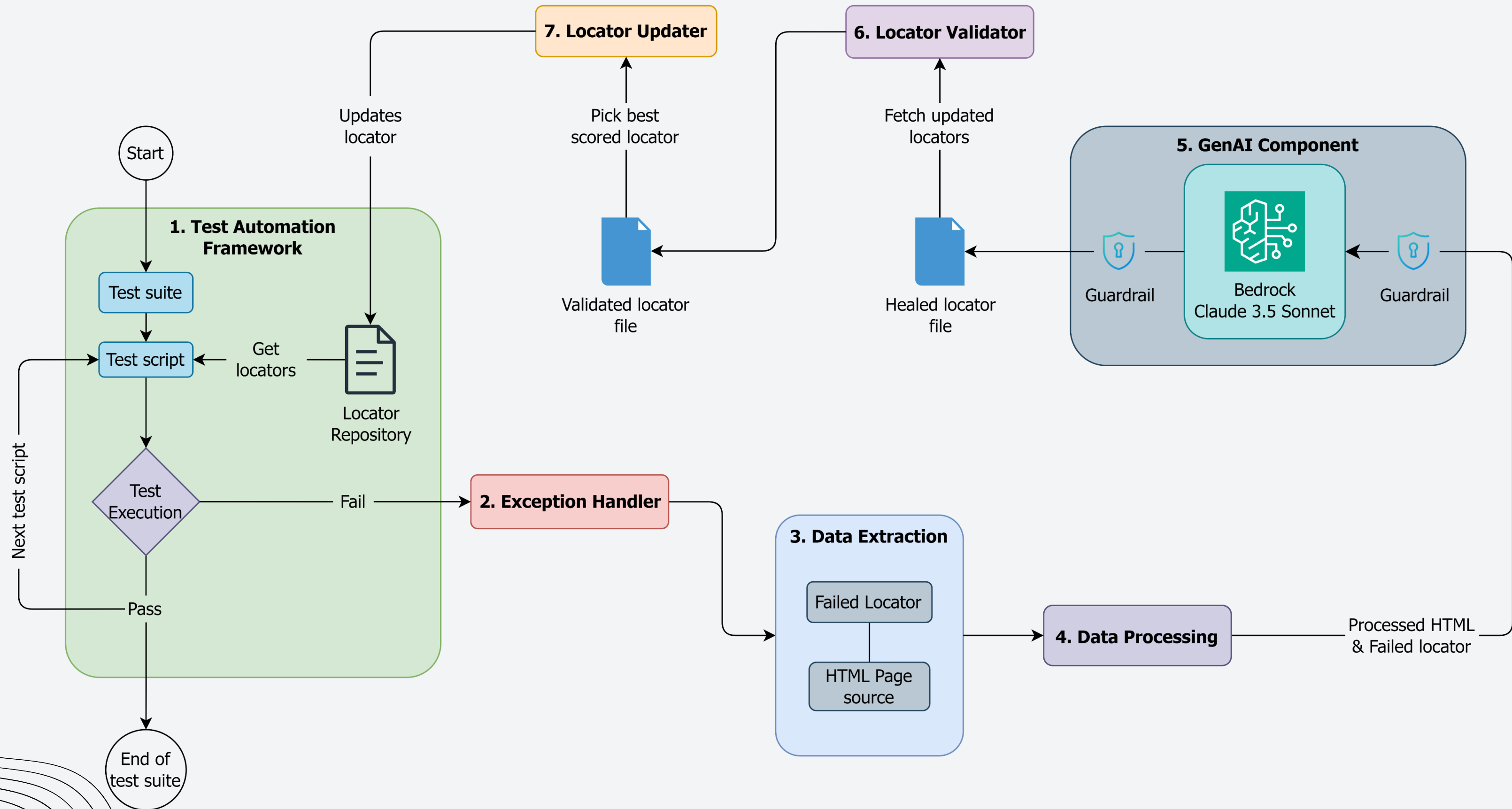
Implementing a GenAI solution to automatically identify and heal failed locators, reducing manual intervention.

Reporting

Create concise reports summarizing the healing process and locator changes, enhancing visibility and tracking.



HIGH-LEVEL ARCHITECTURE DESIGN





DEMO



APPENDIX

KEY CHALLENGES AND OVERCOMES

Here we have outlined **three main challenges** encountered during the implementation which involved HTML processing and LLM (Language Model) integration. Each challenge is paired with the solution implemented to overcome it, showcasing problem-solving and iterative improvement in the development process.

Initially, extracting and processing the HTML was complicated. We resolved this by segregating the failed locators, enabling effective truncation and processing of the HTML.

Processing HTML

The number of tokens sent to the LLM was too high. We optimized both the HTML and the prompts to reduce token usage significantly.

Input to LLM

Early outcomes from the LLM were less accurate and inconsistent. By fine-tuning the Temperature and Top-p parameters, we improved the accuracy and consistency of locator identification.

LLM Output

LLMs ANALYZED

* Based on our research and analysis, Claude 3.5 Sonnet is the preferred model for the use case.

Model	Quality Index (as per Artificial Analysis)	Performance (Output)	Context window
Claude 3 Haiku	54	122 tokens/sec	200K
Claude 3.5 Sonnet *	77	53 tokens/sec	200K
Llama 3.2 11B Vision	53	42 tokens/sec	130K

Table values evaluated from – [AI Model & API Providers Analysis | Artificial Analysis](#)

GUARDRAILS

01

Prompt & Response filter

Detect and block harmful user inputs and model responses.

02

Prompt attacks filter

Detect and block user inputs attempting to override system instructions.

03

Profanity filter

Block profane words in user inputs and model responses.

04

PII filter

Handle data related to privacy, filters the PII such as Name, Password, SSN etc. and masks them while responding.

FUTURE ENHANCEMENTS



By transforming this solution into a Software as a Service (SaaS) model, we can expand its compatibility to support various frameworks.

Support multiple frameworks



Implement retrieval mechanism which automatically understands the change in locator during runtime and executes the step using the updated locator.

Retry Mechanism



Incorporate additional locator strategies and fix dynamic locators. And also improve HTML processing method.

Locator Strategies

THANK YOU

From,
Byte Builders

