



Suhas Bhairav

AI Software UI Translation Text Layout Checker

**Automated checks ensuring translated UI text fits
and never breaks interfaces**

Deliver a repeatable AI-assisted flow to validate
translated UI strings in code, design, and QA.

Suhas Bhairav

suhasbhairav.com

Scope and Automation

Define what to automate: automated UI text layout checks for translations that may break interfaces. Identify what requires human review and what can be auto-approved.

Key move Pilot on one screen; measure layout pass rate before expanding.

PRACTICAL CHECKLIST

- 01** Automate detection of overly long translated words that cause overflow, wrapping, or truncation in UI components.
- 02** Extract strings from source code and design tokens to build a verifiable test corpus.
- 03** Flag font/spacing edge cases and responsive layouts across breakpoints without blocking builds.
- 04** Involve product and design for human reviews of tricky translations or brand constraints.
- 05** Record data requirements: source language, target languages, font metrics, and CSS constraints.
- 06** Define ownership: who approves changes, who monitors failures, and how issues are reported.

Suhas Bhairav AI Resource

Deliver a repeatable AI-assisted flow to validate translated UI strings in code, design, and QA.

Architecture and Data

Outline the AI flow, data sources, and toolchain. Define what to automate versus what to review.

Key move Clarify who approves changes and how reviews are tracked.

PRACTICAL CHECKLIST

- 01** Source UI strings via i18n assets or design tokens, exported as JSON/PO files.
- 02** Use an AI agent (Claude, ChatGPT) to suggest wrap length checks and font fallbacks.
- 03** Integrate with CI: run text checks on pull requests and build pipelines.
- 04** Define rules: max characters per language, line breaks, and padding constraints.
- 05** Maintain human-reviewed exceptions for brand terms and new languages.
- 06** Store audit logs for translations, decisions, and reviewer actions.
- 07** Data needs: font metrics, viewport sizes, and asset mapping.

Suhas Bhairav AI Resource

Deliver a repeatable AI-assisted flow to validate translated UI strings in code, design, and QA.

Adoption Plan & Governance

Provide a four-week, weekly increment plan for teams to adopt the checker. Define success metrics and governance.

Key move Set up a 15-minute weekly review to stay aligned.

PRACTICAL CHECKLIST

- 01** Week 1: inventory all translated UI strings and map to components.
- 02** Week 2: wire automated length checks into CI for one app.
- 03** Week 3: run human reviews for critical strings and brand terms.
- 04** Week 4: expand to two apps and measure defect rate reduction.
- 05** KPIs: layout pass rate, average review time, defect leakage, and contributor velocity.
- 06** Guardrails: never auto-approve critical translations; require human consent.
- 07** Ownership: engineering product manager oversees data quality and releases.

Suhas Bhairav AI Resource

Deliver a repeatable AI-assisted flow to validate translated UI strings in code, design, and QA.

ABOUT THE CREATOR

Suhas Bhairav

I build AI systems that combine distributed architecture, retrieval, knowledge graphs, security, and practical workflow design. My focus is AI that can be used repeatedly by real teams: understandable, reliable, and grounded in the work people already do.

EXPERTISE

- Principal systems architecture and applied AI engineering
- Production-grade AI systems, agentic workflows, RAG, and knowledge graphs
- Full-stack workflow applications across AWS, Google Cloud, Azure, on-premise, and hybrid environments
- Practical AI implementation patterns for automotive, manufacturing, finance, logistics, cybersecurity, sales, and customer operations
- Security, governance, observability, approvals, and failure-mode thinking for operational AI systems

WEBSITE

suhasbhairav.com

EMAIL

suhasbhairav@gmail.com