Technische Hochschule Ingolstadt

UI Fidelity Checker

Concept Design

Ece Sutanrikulu

Lecturer: Prof. Dr. Ignacio Alvar

UXDM_VBC - WS25/26

Problem and context

The Design-Implementation Gap Problem

Problem: Design mockups rarely match pixel-perfect with implementation

Current Pain Points:

1. Manual visual QA is time-consuming and inconsistent
2. Viewport mismatches cause false positives (1920x1080 design vs 1440x900 screenshot)
3. No automated prioritization - everything looks equally important
4. Creating bug tickets manually from screenshots is tedious
5. Existing tools (Percy, Chromatic) require SaaS subscriptions and code integration

Real-world scenario:
UX designer hands off Figma mockup → Developer implements → Needs 3-4 review cycles to catch subtle color/spacing differences → 2 hours wasted per screen

Target users

Primary Users:

- **UX/UI Designers** - QA-ing developer implementations against mockups
- **Frontend Developers** - Self-reviewing before handoff
- **QA Engineers** - Visual regression testing in CI/CD
- **Students/Researchers** - Learning visual testing workflows

Implemented Tool

UI Fidelity Checker - Screenshot-Based Visual Regression

Core Value Proposition:
Compare design mockups with live implementations in one click, get prioritized mismatch reports, and generate ready-to-use tickets.

Key Features:

1. **Dual Input:** Upload both images OR enter URL for auto-screenshot
2. **Smart Viewport Matching:** Auto-uses design dimensions for screenshots
3. **Intelligent Normalization:** 3 sizing modes eliminate false positives
4. **AI-Powered Categorization:** Color, Typography, Spacing, Component State
5. **Priority Ranking:** High/Medium/Low based on visual impact
6. **One-Click Tickets:** 4 formats (Markdown, GitHub, Jira, JSON)

Visual: Screenshot of main interface showing dual upload areas

Value: faster handoff, clearer tickets

Primary user flow

Flow:
Designer exports homepage mockup (1920x1080 PNG)
↓
Opens UI Fidelity Checker
↓
Uploads mockup + enters live URL/Screenshot
↓
Tool captures screenshot at design dimensions
↓
Automatic comparison with prioritized mismatches
↓
One-click ticket creation for GitHub Issues

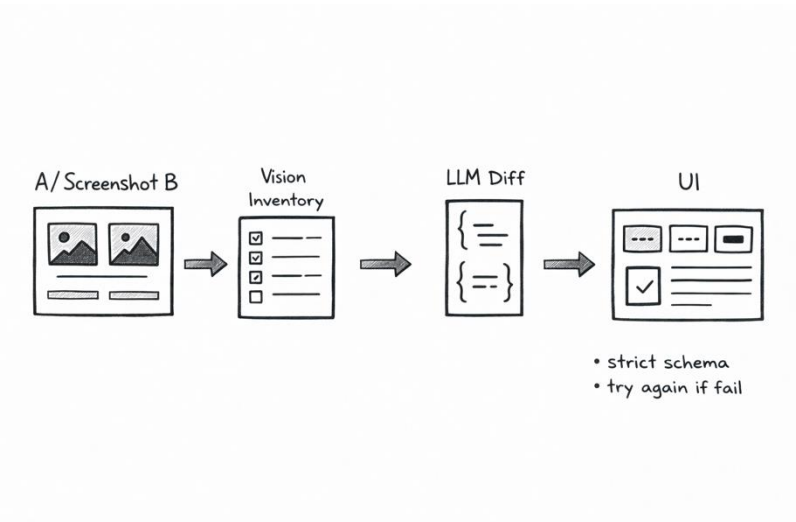
AI / Vibecoding angle

AI-Assisted Development (Claude/Cursor):

1. Entire codebase built with AI pair programming
2. Complex algorithms (viewport normalization, diff generation) designed through conversation
3. 12-step implementation plan generated by AI
4. Unit tests written and debugged with AI assistance

Heuristic "AI" in Product:

1. **Smart Categorization:** Pattern matching for mismatch types
2. **Priority Ranking:** Area + intensity scoring algorithm
3. **Viewport Mismatch Detection:** Ratio analysis with threshold intelligence



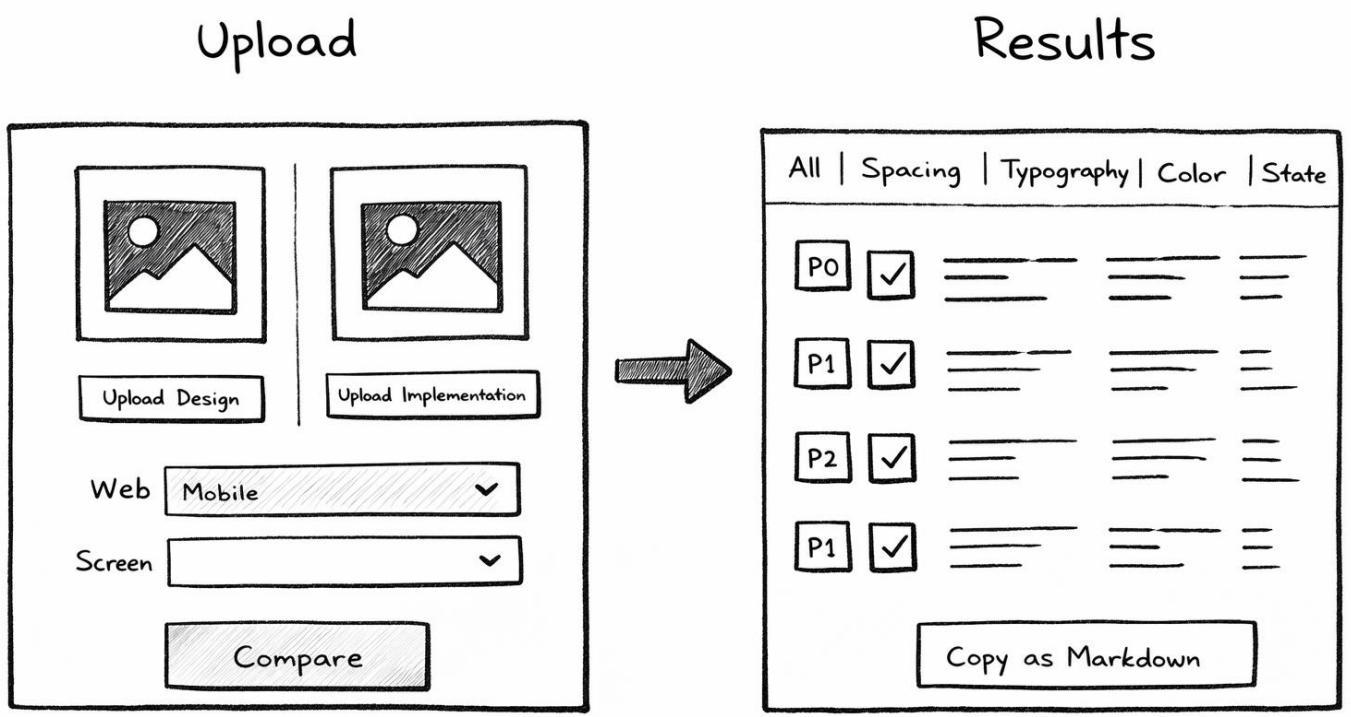
Tech Stack and Features

Tech Stack:
Frontend: Next.js 16.1.3 (App Router), React 19, TypeScript 5.9
Styling: Tailwind CSS 4.x, Custom CSS animations
Image Processing: Sharp (normalization), Pixelmatch (diffing)
Automation: Playwright (Chromium screenshots)
Testing: Custom tsx test runner (21 unit tests, all passing)

Features Implemented:

1. Dual input modes (upload + URL)
2. Viewport & size normalization (3 modes)
3. Automatic dimension extraction
4. Viewport mismatch detection & warnings
5. Smart categorization (4 types)
6. Priority ranking (3 levels)
7. Interactive mismatch selection
8. Ticket generation (4 formats)
9. One-click export (copy/download/GitHub)
10. Responsive UI with "Precision Studio" aesthetic

Rough UX sketch / Wireframe



Live Demo


UI
FIDELITY
Pixel-perfect visual regression analysis.
Find every mismatch.
SIMILARITY SCORE
DESIGN MOCKUP
DROP OR CLICK
IMPLEMENTATION
UPLOAD
URL
DROP OR CLICK

Technische Hochschule Ingolstadt

Thanks!