



Group 8H

22125025 - Huynh Huu Hau

22125037 - Duong Ngoc Quang Khiem

22125054 - Cao Vo Nhat Minh

22125093 - Tran Nhat Thanh

Table of contents

01

GUI

03

UX

02

**BROWSER &
LINK**

04

**“AI-FIRST”
APPROACH**

Software under test (SUT)

The screenshot shows a web application for managing a tool shop. At the top, there's a navigation bar with links for 'Home', 'Categories', 'Contact', 'Sign in', and language selection ('EN'). Below the navigation is a large image of various tools hanging on a wooden pegboard. On the left, there's a sidebar with a 'Sort' dropdown, a 'Price Range' slider from 1 to 200, and a 'Search' input field with a button. The main content area displays three tool products: 'Combination Pliers' (yellow), 'Pliers' (red), and 'Bolt Cutters' (red). Each product has a small image, a title, and a color-coded status indicator (green for Combination Pliers, orange for Pliers and Bolt Cutters).

Sprint 5

Sprint 5

This screenshot shows the same web application after a bug has been introduced. The interface is identical to the first one, but now features several red 'DANGER' icons overlaid on the tool images. These icons are placed on the handles of the pliers and bolt cutters, indicating a critical issue or bug that needs attention. The rest of the page, including the sidebar and search functionality, remains the same.

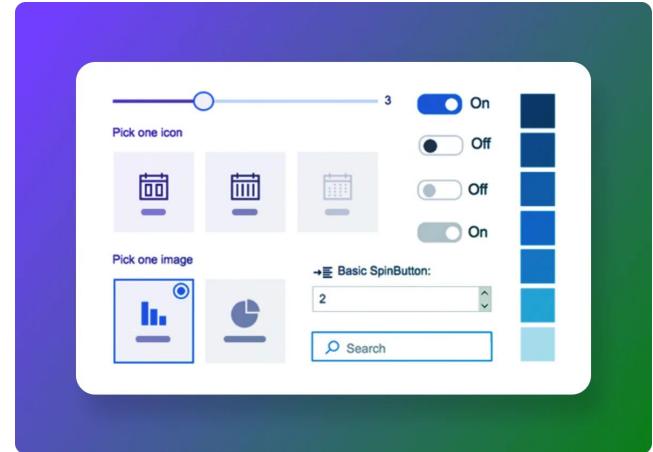


01 GUI Testing

What is GUI testing?

GUI testing is the process of examining everything a user sees and interacts with on a screen.

We check elements like **buttons, menus, input fields, images, icons, visual design,...** to ensure they work as intended.



UI vs. GUI

What's the relationship?

UI (User Interface)

- All interaction methods.
 - Includes **voice, gesture, command-line, touch, physical buttons**.

GUI (Graphical User Interface)

- A type of UI that uses **graphics**: buttons, icons, colors, windows, menus.
 - What we see in **web apps, mobile apps, desktop apps**



```
create .gitignore-format
create .editorconfig
create .eslintrc
create .angularcli-clipboard-app-component
create src/app/angular-clipboard/app/component
create src/app/angular-clipboard/app/component
create src/app/angular-clipboard/app/component
create src/app/environment.ts
create src/app/index.tsx
create src/app/index.ts
create src/favicon.ico
create src/icon.png
create src/icon.svg
create src/main.ts
create src/main.ts.map
create src/main.ts.webapp
create tsconfig.eslint.json
create tsconfig.json
create tsconfig.lib.json
create tsconfig.spec.json
create tslint.json
create angular-cli.json
create config/environments/dev.ts
```

Why is GUI Testing Important?



Why is GUI Testing Important?

User-Centric Focus:

The UI is the main point of interaction, so its **reliability** is crucial for a positive user experience.

Example: A payment button is poorly placed or hard to find.



Users abandon their shopping cart, leading to lost sales.

Find & Fix Defects:

The primary goal is to **find visual defects**, making the app appealing and user-friendly.

Example: On a specific phone, the product's description text overlaps the product's image.



The page looks broken and unprofessional.

Why is GUI Testing Important?

Ensure Visual Consistency:

We must verify the software looks the same **across different devices and platforms.**

Example: A page looks fine on Chrome but misaligned on Firefox or Safari.



Hundreds of cross-browser complaints.

Protect Brand Reputation:

A polished, bug-free UI **enhances brand reputation** and helps detect bugs early, which reduces fixing costs.

Example: A banking app shows clipped text on the homepage.



Users lose trust:
“If they can’t fix the UI, can they protect my money?”

How We Test: The GUI Checklist

A **GUI Checklist** is a predefined list of criteria used to validate the interface.

It helps us systematically check elements. We've organized our checklist into 4 main categories following the template provided:

- 1. Colors** (*Checking for consistency and proper design*)
- 2. Content** (*Checking fonts, alignment, spacing, and spelling*)
- 3. Images** (*Checking for correct display, alignment, and visual consistency*)
- 4. Form** (*Checking labels, input fields, and states*)

GUI Checklist - Colors

1.1 COLORS

- 1.1.1 Is the website's background color dark?
- 1.1.2 Does the website's background color confuse or overwhelm users?
- 1.1.3 Are the colors between different sections clearly differentiated and consistent?
- 1.1.4 Is the text color for normal content consistent?
 - Do the text colors for bold, italic, and hyperlinks differ clearly and stand out from normal text?
- 1.1.6 Does the text color of visited links change appropriately?
- 1.1.7 Does the text color of links change on hover?
- 1.1.8 Is the text color inside textboxes consistent when users input data?
- 1.1.9 Are the background colors of button groups consistent with each other?
- 1.1.10 Is the text color on buttons consistent?
 - Are the colors of disabled input fields consistent and clearly different from other input fields?
- 1.1.11 input fields?

GUI Checklist - Content

1.2 CONTENT	
	Is the font used throughout the website consistent across components (buttons, textboxes, content, headings, links...)?
1.2.1	Is the font size for normal text correct according to basic readability standards?
1.2.2	Is the font size displayed correctly on buttons and textboxes?
1.2.3	Are heading font sizes larger and more prominent than normal text?
1.2.4	Is the website logo placed at the top-left corner?
1.2.5	Is all text aligned correctly (left, right, center) depending on the design?
1.2.6	Are all headings aligned properly?
1.2.7	Is the overall website layout clear and logical?
1.2.8	Does each webpage display a clear title?
1.2.9	Is the search bar prominently displayed and easy to find?
1.2.10	Is the product listing section displayed consistently?
1.2.11	Are products of the same category grouped and displayed appropriately?
1.2.12	Are all error message texts spelled correctly on this screen?
1.2.13	Are all tooltip/help texts spelled correctly on this screen?
1.2.14	Are all UI component texts (buttons, menus, labels...) spelled correctly?
1.2.15	

GUI Checklist - Images & Form

1.3 IMAGES

- 1.3.1 Are all essential images displayed correctly?
- 1.3.2 Are all images aligned correctly?
- 1.3.3 Are banner images displayed correctly and consistently across different window sizes?
- 1.3.4 Is text wrapping around images displayed correctly?
- 1.3.5 Does the website remain visually readable even when images fail to load?

1.4 FORM

- 1.4.1 Do required and optional fields have clear indicators?
- 1.4.2 Do complex input fields display input instructions or constraints?
- 1.4.3 Are default values (if any) displayed correctly when the page loads or reloads?
 - Are all parts of a table or form displayed correctly, and are selected values shown in the correct position?
- 1.4.4
- 1.4.5 Can only one option be selected within a radio button group?
- 1.4.6 Can users select one or multiple options in checkboxes?
- 1.4.7 Is the information typed into password fields hidden properly?

SELENIUM



Selenium is an automation framework that controls a web browser just like a real user.

It allows testers to simulate actions such as clicking, typing, scrolling, and checking UI properties.

For this seminar,

We are focusing on using Selenium to demo some tests about the visual aspects of the GUI (layout, visuals, color, and position).

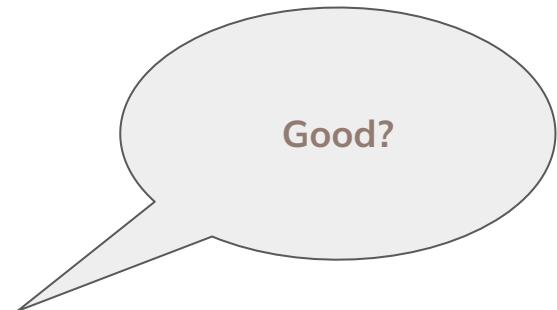
SELENIUM

- ✓ **Free & Open Source**

No license, no subscription, no restrictions — anyone can use it.

- ✓ **Automates Repetitive GUI Checks**

Instead of manually checking every product card, button, or label, Selenium can repeat the same UI checks hundreds of times without mistakes.



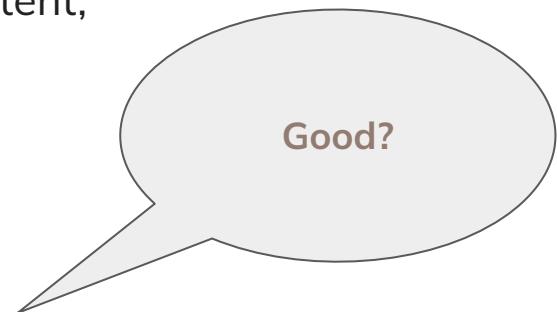
SELENIUM

✓ Reads UI Properties Directly

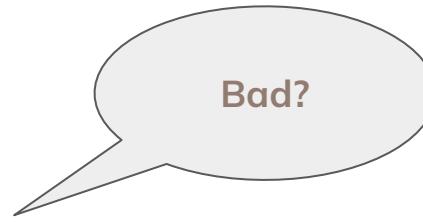
Selenium can extract CSS values such as color, font-size, spacing, margin, position, and visibility, letting us verify UI consistency automatically.

✓ Basic but Very Practical

Selenium isn't "AI smart," but it's extremely reliable for consistent, rule-based UI testing.



SELENIUM



Bad?

High Maintenance: Test scripts are "brittle". If a developer changes the UI, the script will often break and must be manually updated by the tester.

No Built-in Image/Visual Testing: By default, Selenium cannot see and feel. It cannot tell whether the text color is readable on the background. It can only read the CSS code.



Selenium can be extended with visual testing tools and plugins, making it flexible and highly effective for GUI testing.

DEMO

Bug Hunting Mode - Find & Report Bugs! Bug Hunting Guide

Home Categories Kontakt Sign in



Sort by

Price Range

1 100 200

Search

X Search



Combination Pliers

CO2: A B C D E

€11.15



Pliers

CO2: A B C D E

€17.01



Bolt Cutters

CO2: A B C D E

€10.01

DEMO 1 - Spelling Check

Goal: To automatically find both visible and hidden text content for spelling errors that damage brand reputation and user trust.

Checklist:

- 1.2.15 (visible): Are all UI component texts (buttons, menus, labels) spelled correctly?
- 1.2.14 (hidden): Are all tooltip/help texts spelled correctly?



Claw Hammer with Shock Reduction Grip

CO₂: A B C D E

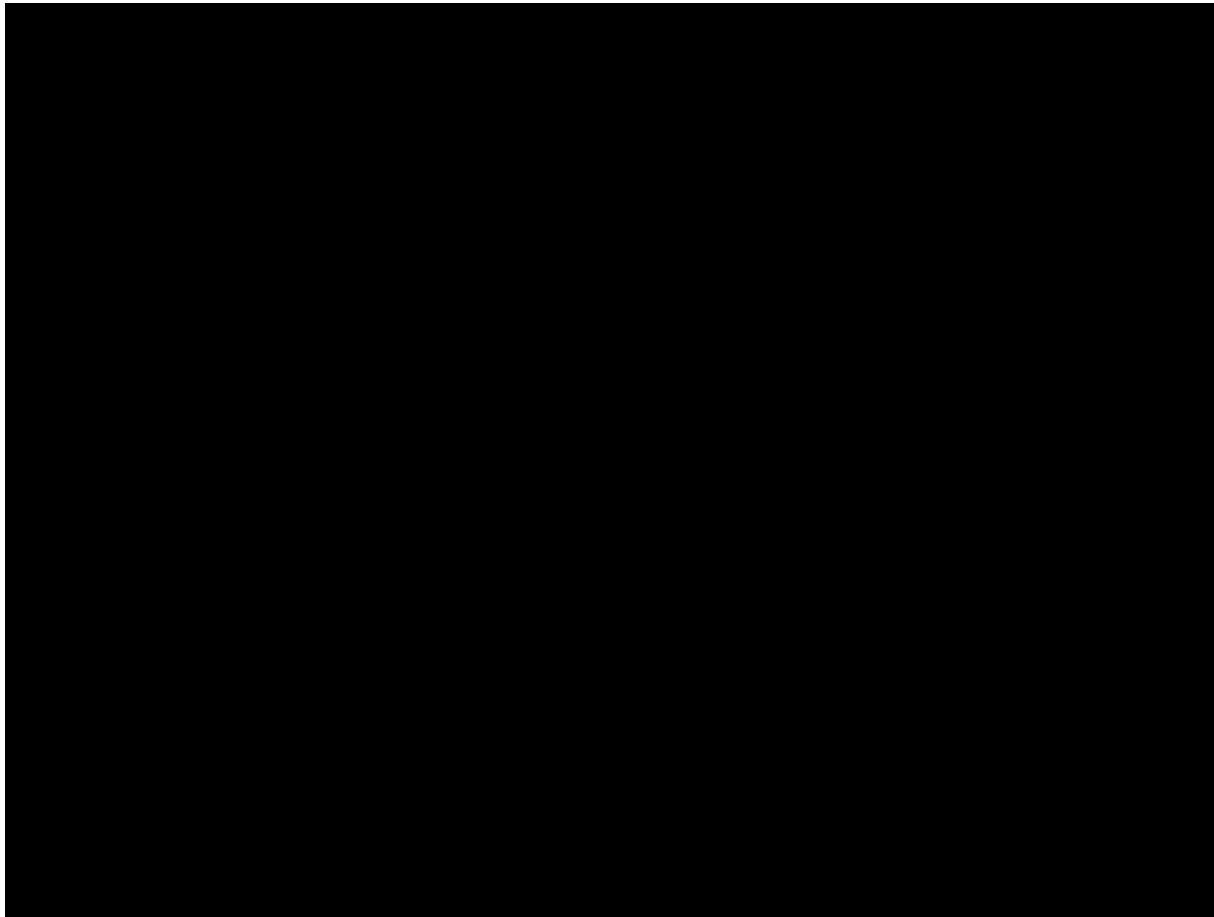
A = most environmentally friendly, E = higher environmental impact

DEMO 1 - Spelling Check

Selenium + Python library *pyspellchecker*.

Why use Automation?

- Hidden Text: Manual testers often forget to hover over every icon to check tooltips.
- Human Eye: Our brains auto-correct typos (e.g., reading "Contakt" as "Contact").
- Solution: Selenium reads the raw code and catches errors.



DEMO 2 - Color Consistency

Goal: To automatically audit the color consistency of specific UI components (CO2 Badges & Pagination) across multiple pages.

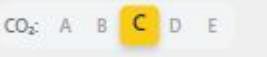
Checklist:

- [1.1.3](#): Are section colors clearly differentiated and consistent with the design?
- [1.1.9](#): Are the background colors of UI components groups (buttons) consistent with each other?

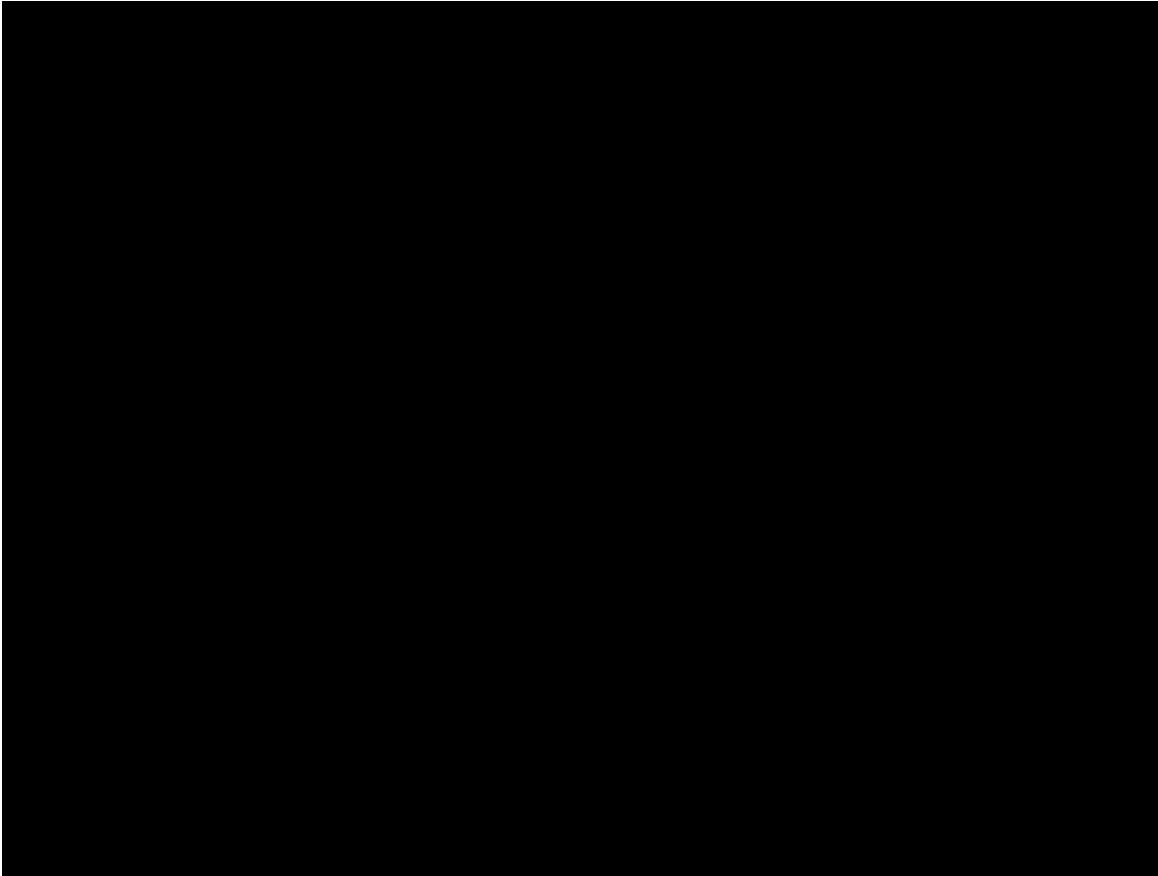
DEMO 2 - Color Consistency

It specifically targets:

- CO2 Badges: Verifying that each rating (A, B, C, D, E) has its own unique, consistent color on every product card.
- Pagination: Verifying that the "Active" page number always uses the standard blue background.



Imagine this page has
more than 200
products?



DEMO 3 - Layout & Image Check

Goal: To use Selenium to scan a list of components and find complex layout, consistency, and image bugs in Mobile Emulation mode then compare with Desktop mode.

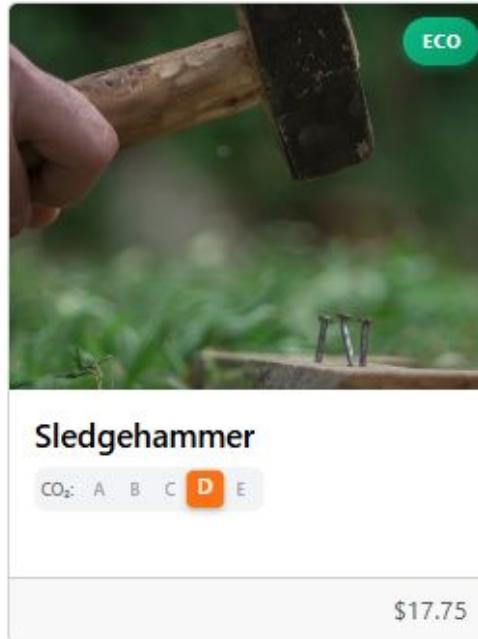
Checklist:

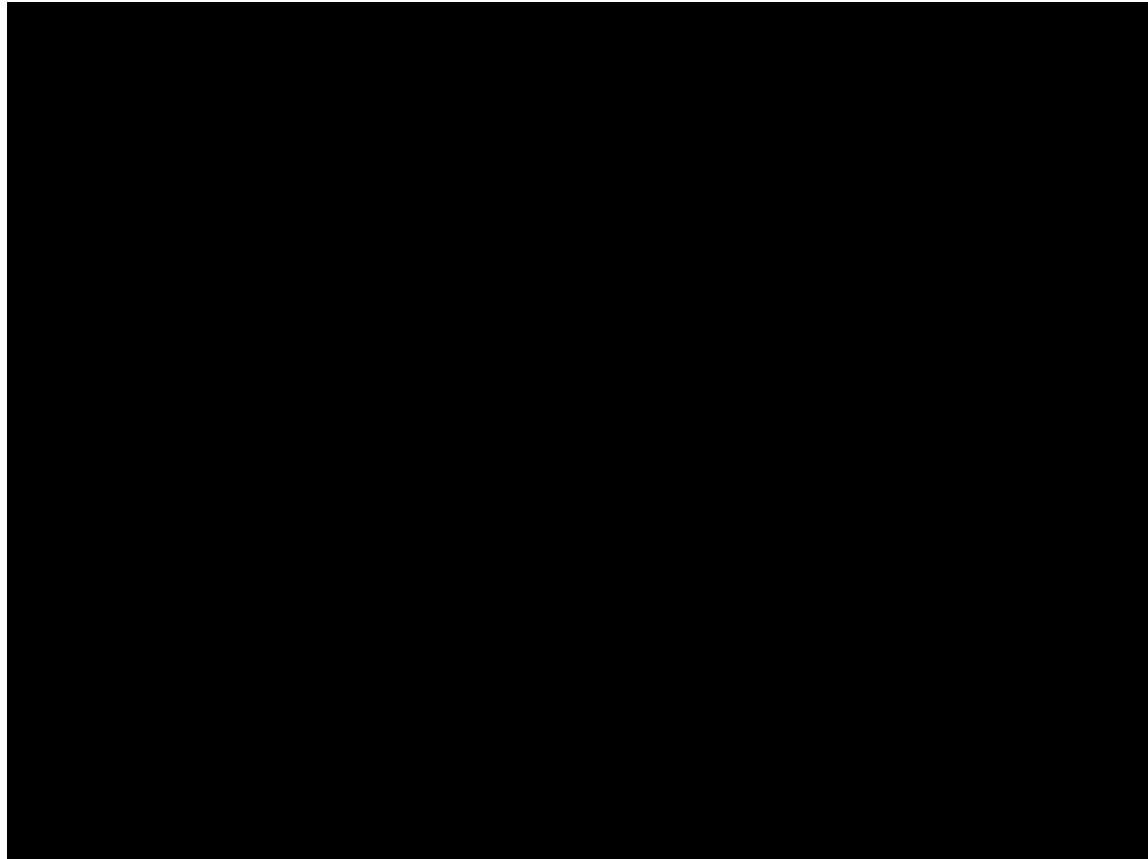
- [1.2.11](#): Is the product listing section displayed consistently?
- [1.3.1](#): Are all essential images (*product's image*) displayed correctly?
- [1.2.6](#): Is all text aligned correctly (left, right, center) depending on the design?

This script is configured to run in Mobile Emulation mode to simulate an iPhone.

DEMO 3 - Layout & Image Check

For each product card, I check the **layout size**, the product **image** visibility, the presence of **title/price/CO₂ badge**, whether **tooltips** exist, and whether the **fonts and alignments** remain consistent across all items.





DEMO 3 - Mobile vs. Desktop

✗ TITLE_ALIGN_INCONSISTENT:

- Page 1 - 'Pliers': text-align is start, violates standard of right.
- Page 1 - 'Long Nose Pliers': text-align is start, violates standard of right.
- Page 1 - 'Slip Joint Pliers': text-align is start, violates standard of right.
- Page 1 - 'Claw Hammer with Shock Reduction Grip': text-align is start, violates standard of right.
- Page 1 - 'Hammer': text-align is start, violates standard of right.
- Page 2 - 'Court Hammer': text-align is start, violates standard of right.
- Page 2 - 'Adjustable Wrench': text-align is start, violates standard of right.
- Page 2 - 'Open-end Spanners (Set)': text-align is start, violates standard of right.
- Page 2 - 'Mini Screwdriver': text-align is start, violates standard of right.
- Page 3 - 'Sheet Sander': text-align is start, violates standard of right.
- Page 3 - 'Cordless Drill 20V': text-align is start, violates standard of right.
- Page 3 - 'Cordless Drill 18V': text-align is start, violates standard of right.

✗ CARD_HEIGHT_INCONSISTENT:

- Page 1 - Product #6: height is 405px, violates standard of 381px. (Caused by long title?)
- Page 2 - Product #2: height is 405px, violates standard of 381px. (Caused by long title?)

Mobile result

✗ TITLE_ALIGN_INCONSISTENT:

- Page 1 - 'Bolt Cutters': text-align is right, violates standard of start.
- Page 1 - 'Slip Joint Pliers': text-align is right, violates standard of start.
- Page 1 - 'Claw Hammer with Shock Reduction Grip': text-align is right, violates standard of start.
- Page 1 - 'Claw Hammer': text-align is right, violates standard of start.
- Page 2 - 'Claw Hammer with Fiberglass Handle': text-align is right, violates standard of start.
- Page 2 - 'Court Hammer': text-align is right, violates standard of start.
- Page 2 - 'Mini Screwdriver': text-align is right, violates standard of start.
- Page 3 - 'Sheet Sander': text-align is right, violates standard of start.
- Page 3 - 'Belt Sander': text-align is right, violates standard of start.

✗ CARD_HEIGHT_INCONSISTENT:

- Page 1 - Product #4: height is 399px, violates standard of 381px. (Caused by long title?)
- Page 1 - Product #5: height is 399px, violates standard of 381px. (Caused by long title?)
- Page 1 - Product #6: height is 405px, violates standard of 381px. (Caused by long title?)
- Page 2 - Product #1: height is 399px, violates standard of 381px. (Caused by long title?)
- Page 2 - Product #2: height is 405px, violates standard of 381px. (Caused by long title?)
- Page 2 - Product #3: height is 397px, violates standard of 381px. (Caused by long title?)

✗ IMAGE_NOT_VISIBLE:

- Page 1 - Product #5: Image is hidden (d-none).
- Page 2 - Product #4: Image is hidden (d-none).
- Page 3 - Product #2: Image is hidden (d-none).

Desktop result



02

Browser Compatibility & Link Testing





Browser Compatibility

Ensuring a consistent user experience for everyone, everywhere.



What is Browser Compatibility Testing?

WHAT?

Testing a web app across all browsers (Chrome, Firefox, Safari), operating systems (Windows, macOS, iOS), and devices (desktops, tablets, phones).

WHY?

Because browsers interpret code (HTML, CSS, JS) differently.

GOAL

To ensure a consistent, functional, and usable experience for ALL users.

Our Two-Part Testing Strategy

Local Playwright

- Test 3 Core Engines + 3 Page Layouts.
- Fast, free, and efficient. Catches 90% of bugs.



BrowserStack

- Test on Real Operating Systems + Real Hardware.
- Catches the last 10% of tricky, device-specific bugs.

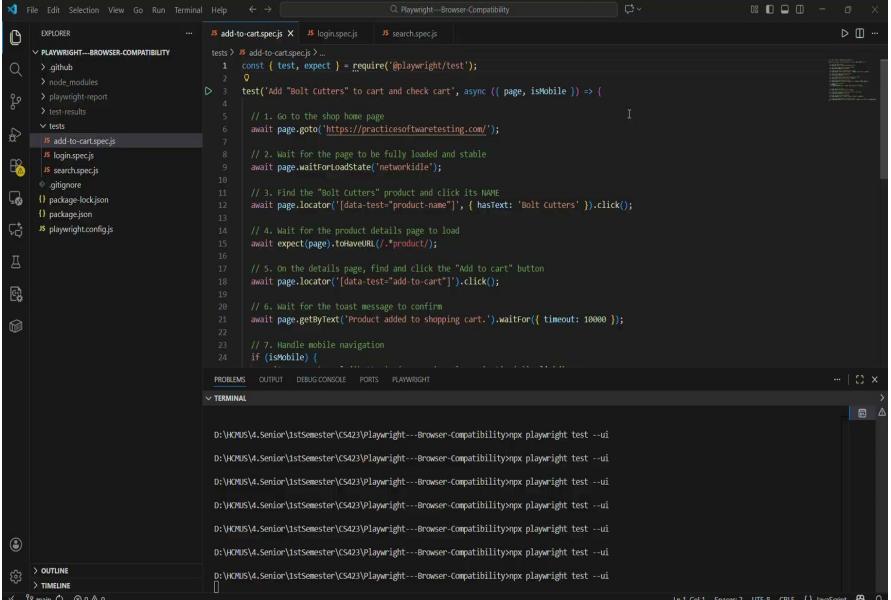


Approach 1: Local Playwright (Automation)

What is it? An automation library where we write code to run tests automatically. It's fast and repeatable.

Demo Scope: We automated 3 core tasks:

1. User Login
2. Searching for an Item
3. Adding an Item to the Cart



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PLAYWRIGHT-BROWSER-COMPATIBILITY". The "tests" folder contains three files: "add-to-cart.spec.js", "login.spec.js", and "search.spec.js".
- Code Editor:** The active file is "add-to-cart.spec.js". The code is a Playwright test script:

```
// 1. Go to the shop home page
await page.goto('https://practicesoftwaretesting.com/');

// 2. Wait for the page to be fully loaded and stable
await page.waitForLoadState('networkidle');

// 3. Find the "Bolt Cutters" product and click its NAME
await page.locator('[data-test="product-name"]').click();

// 4. Wait for the product details page to load
await expect(page).toHaveURL('/#/product');

// 5. On the details page, find and click the "Add to cart" button
await page.locator('[data-test="add-to-cart"]').click();

// 6. Wait for the toast message to confirm
await page.getByText('Product added to shopping cart.').waitFor({ timeout: 10000 });

// 7. Handle mobile navigation
if (!isMobile) {
```

- Terminal:** Shows multiple command-line entries of the command "playwright test --ui" being run in different browser environments (chromium,firefox,webkit,edge).
- Status Bar:** Shows "In 1 Col 1" and "JavaScript" as the current file type.

Why These 6 Setups?

The 3 Engines:

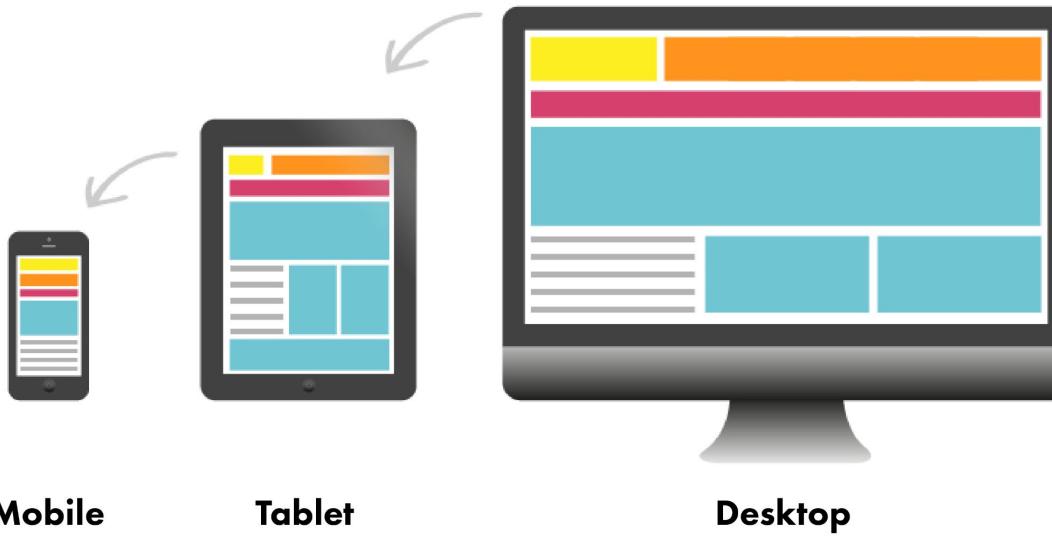
- **Chromium** (for Chrome, Opera, Edge)
- **Firefox** (Gecko)
- **WebKit** (for all Safari browsers)



Why These 6 Setups?

The 3 Form Factors:

- **Desktop** (Wide layout, full menus)
- **Tablet** (Medium layout)
- **Mobile** (Narrow "hamburger" layout)



Approach 2: BrowserStack (Manual)

What is it?

A cloud platform providing access to 3000+ real browsers and devices for testing. No setup or installation required, you test live in their cloud.

Free Plan Limit

The free plan gives access to live testing, but with a major limitation: a **1-minute session timer** per device.

Demo Scope

We ran 5 manual tests:

- Mac (Chrome, Firefox, Safari)
- Windows (Chrome, Firefox, Edge)
- iPhone & iPad (Safari)
- Android (Chrome)

BrowserStack Demo Videos

The screenshot shows the BrowserStack Live dashboard. On the left, there's a sidebar with various icons for different testing tools and features. The main area displays a grid of device icons representing different combinations of operating systems (iOS, Android, Windows, Mac) and browser versions (Safari, Chrome, Firefox, Edge, Opera, Internet Explorer). A tooltip indicates that each device is available for up to 1 minute during the free trial. A prominent call-to-action button says "Get Extension →". To the right of the grid, there's a section for "Favourites" which lists 10 out of 12 devices. The "Sonoma" entry is highlighted. At the bottom right, there's a "Talk to an Expert" button.

See how 20+ AI Agents can make your QA faster and smarter. Join 10k+ QAs at global leadership summit! Register Now

BrowserStack Live App Live Automate Accessibility Testing Products

Invite team Plans and pricing ? 📡 🔔 🔍 Buy a plan

Each device is available for up to 1 minute during Free Trial. For full access: Buy Pro plan

Launch Live session directly from your webpage with one-click through Testing Toolkit extension. Rated 5/5 on Chrome store

Get Extension →

Favourites (10/12)

Device	Version	Platform
Tahoe	26 latest	iOS
Sequoia	144 latest	Android
Sonoma	145 beta	Windows
Ventura	143	Mac
Monterey	142	
Big Sur	141	
Catalina	140	
Mojave	139	
High Sierra	138	
Sierra	137	
El Capitan	136	

55 more 59 more 83 more 49 more

Talk to an Expert

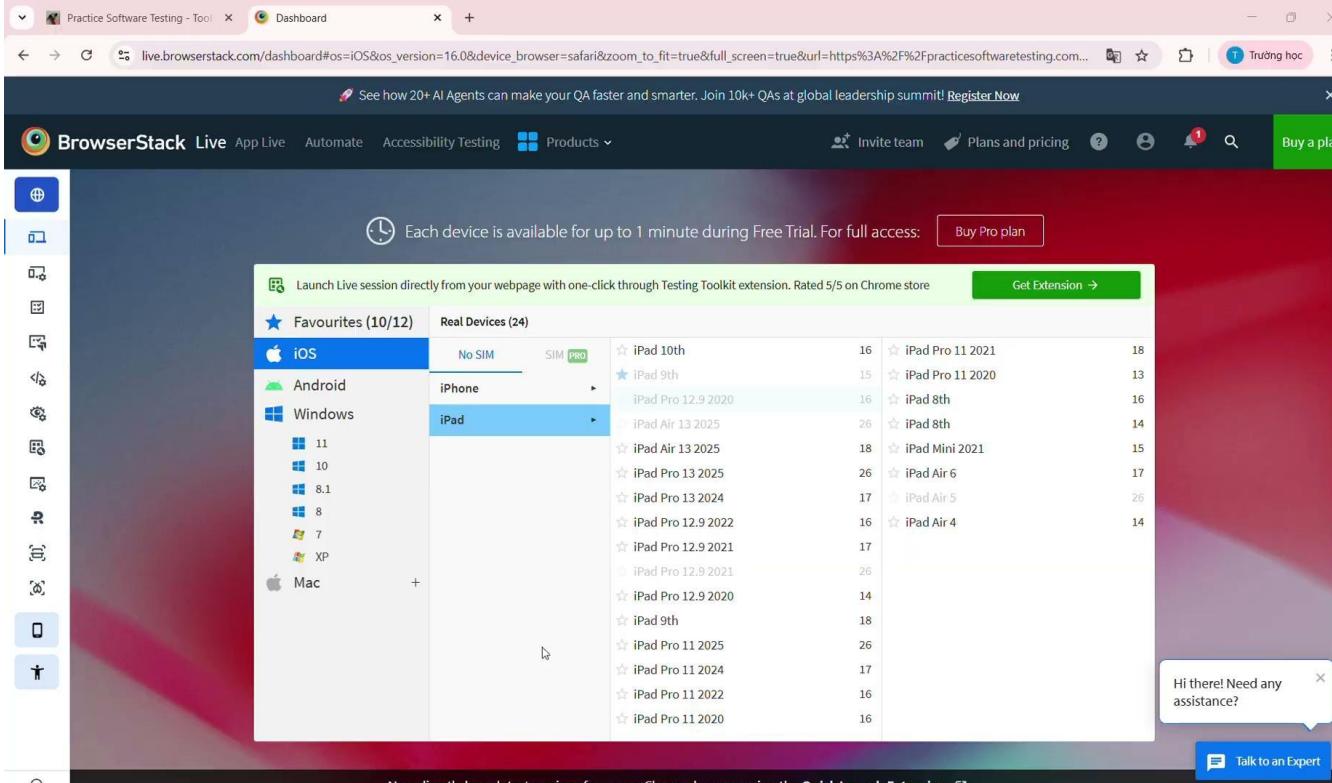
Macbook

BrowserStack Demo Videos

The screenshot shows the BrowserStack Live dashboard. At the top, there's a banner with a rocket icon and the text: "See how 20+ AI Agents can make your QA faster and smarter. Join 10k+ QAs at global leadership summit! Register Now". Below the banner, the main navigation bar includes "BrowserStack Live", "App Live", "Automate", "Accessibility Testing", "Products", "Invite team", "Plans and pricing", and "Buy a plan". A sidebar on the left contains various icons for device testing, such as mobile phones, tablets, desktops, and accessibility tools. The central part of the screen displays a grid of browser versions for Windows. The grid has seven columns and multiple rows. The columns are labeled with browser icons: Internet Explorer (IE), Firefox (Fx), Google Chrome (Cr), Opera (Op), Yandex (Y), and Microsoft Edge (Ed). The rows represent different versions of Windows, starting from 11 at the top and going down to XP at the bottom. Each row contains several browser versions, with some being marked as "latest" or "beta". For example, the first row (Windows 11) shows IE 142 latest, Fx 144 latest, Cr 142 latest, Op 123 latest, Y 14.12 latest, and Ed 5.1 latest. The second row (Windows 10) shows IE 143 dev, Fx 145 beta, Cr 143 beta, Op 124 dev, Y 122, and Ed 141. The third row (Windows 8.1) shows IE 142 beta, Fx 143, Cr 144 dev, Op 122, Y 121, and Ed 141. The fourth row (Windows 8) shows IE 141, Fx 142, Cr 140, Op 120, Y 119, and Ed 138. The fifth row (Windows 7) shows IE 140, Fx 141, Cr 139, Op 118, Y 117, and Ed 137. The sixth row (Windows XP) shows IE 139, Fx 140, Cr 138, Op 116, Y 117, and Ed 136. The seventh row (Mac) shows IE 136, Fx 137, Cr 136, Op 115, Y 116, and Ed 135. At the bottom of the grid, there are links for "54 more", "104 more", "97 more", and "90 more". A "Buy Pro plan" button is located above the grid. A message at the bottom of the dashboard says: "Each device is available for up to 1 minute during Free Trial. For full access: Buy Pro plan". A "Get Extension →" button is also present. A "Talk to an Expert" button is located in the bottom right corner. The overall background of the dashboard features a blue and white abstract design.

Windows

BrowserStack Demo Videos



Ipad (Tablet)

BrowserStack Demo Videos

The screenshot shows the BrowserStack Live dashboard. On the left, there's a sidebar with various icons for different testing features like mobile, accessibility, and automation. The main area displays a list of devices categorized by platform: iOS, Android, and Windows. Under iOS, there are sections for 'No SIM' and 'SIM PRO'. The 'SIM PRO' section is currently selected and shows a dropdown menu with the option 'iPhone' highlighted. This dropdown also includes the text 'Device is SIM card and Apple Pay enabled'. To the right of the dropdown, there's a list of iPhone models with their respective IDs and star ratings. A green button at the top right of the device list says 'Get Extension →'. At the bottom of the page, there's a banner that reads 'Now directly launch test sessions from your Chrome browser using the Quick Launch Extension'.

Platform	Type	Device	ID	Rating	Count	
iOS	No SIM	iPhone 16	18	★	iPhone 13	16
	No SIM	iPhone Air	26	★	iPhone 14	26
	No SIM	iPhone 17	26.1	★	iPhone 14	18
	No SIM	iPhone 17	26	★	iPhone 13 Pro	15
	No SIM	iPhone 16 Pro	18	★	iPhone 13 Mini	15
	No SIM	iPhone 16 Plus	18	★	iPhone 13 Pro Max	15
	No SIM	iPhone 11	13	★	iPhone 17 Pro Max	26
Android	No SIM	iPhone 15	17	★	iPhone 17 Pro	26
	No SIM	iPhone 15	26	★	iPhone 16 Pro Max	18
	No SIM	iPhone 13 Pro Max	18	★	iPhone 15 Pro Max	26
	No SIM	iPhone 13	15	★	iPhone 15 Pro	17
	No SIM	iPhone 14 Pro	26	★	iPhone 15 Plus	17
	No SIM	iPhone 14 Pro	16	★	iPhone 13	18
	No SIM	iPhone 14 Plus	16	★	iPhone 13	17
	No SIM	iPhone 15 Pro	17	★	iPhone 12	17
	No SIM	iPhone 16	18	★	iPhone 12 Pro	17
	No SIM	iPhone 17	26	★	iPhone 12 Pro Max	18
Windows	No SIM	Windows 11	11	★	Windows 10	10
	No SIM	Windows 10	10	★	Windows 8.1	8.1
	No SIM	Windows 8	8	★	Windows 7	7
	No SIM	Windows XP	XP	★	Windows Vista	Vista
	Mac	+ Add Device				

Iphone

BrowserStack Demo Videos

The screenshot shows the BrowserStack Live dashboard. On the left, there's a sidebar with various icons for managing devices and sessions. The main area displays a list of devices categorized by platform: Favourites (10/12), Real Devices (35), iOS, Android, Windows, and Mac. Under the Android section, 'Samsung' is selected. A modal window is open, showing a list of Samsung devices with their model names and IDs. The modal includes a 'Get Extension →' button. At the bottom of the page, there's a message bubble from a chatbot and a 'Talk to an Expert' button.

Category	Device	ID
Real Devices (35)	Galaxy S23	13
	Galaxy Z Fold 5	13
	Galaxy Z Fold 5	13
Galaxy	S21	12
	S20	10
	S10	9
Galaxy Tab	S11	16
	S7	11
	S7	10
Galaxy A	Tab S10+	15
	A52	11
	A51	10
Galaxy M	A16 5G	15
	M32	11
	M30	13
Galaxy A	A16 5G	15
	A34	13
	A35	14
Galaxy S	A11	10
	S21 Ultra	11
	S21	11
Galaxy A	A10	9
	S21 Ultra	11
	S20	10
Galaxy M	M52	11
	S24	14
	S24 Ultra	14
Galaxy Note	M50	11
	S25	15
	S25 Ultra	15
Galaxy Tab	S8	12
	S7	13
	S9	13

Android

BrowserStack: Why Test Real OS & Hardware?

Main Goal: To catch bugs that **emulators miss**. We test on **Real Operating Systems** and **Real Hardware**.





Link Testing

Ensuring every click leads to the right destination.



What is Link Testing (Link Checking)?

What is it?

- The process of systematically scanning a website to find and verify all hyperlinks.
- This includes: **Internal links** (to your own pages), **External links** (to other sites), and **Anchor links** (on-page jumps).

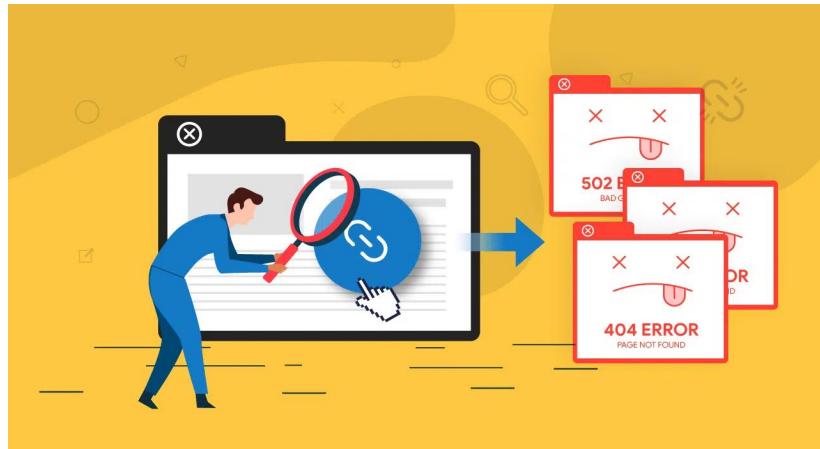
Why do we do it? To find broken links, which most often lead to a 404 "Not Found" error.



Why Care About Broken Links?

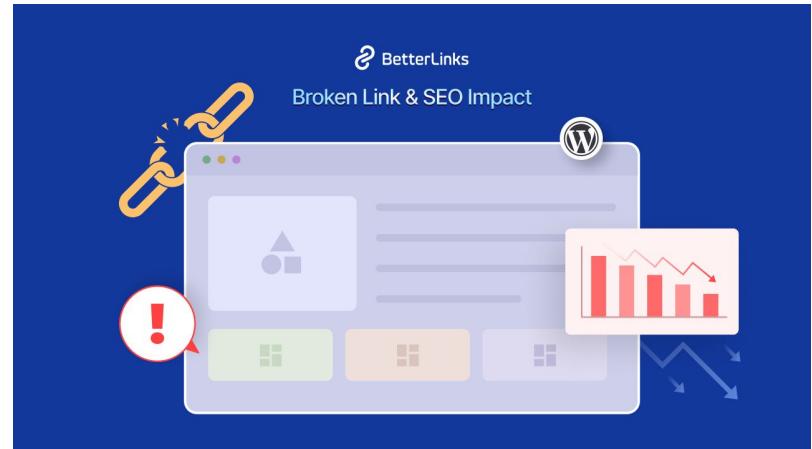
Reason 1: Bad User Experience (UX)

- It frustrates users and stops them from finding the information they need.
- Makes your site seem unprofessional and unreliable.



Reason 2: Bad SEO (Search Engine Optimization)

- Search engines (like Google) see broken links as a sign of a low-quality or poorly maintained site.
- This can lower your search ranking, making your site harder to find.



Approach 1: Simple Scan ("Check My Links")



What is it?

- A free Chrome Extension that scans the *single page* you are currently on.

How it works:

- It instantly crawls the page and highlights all links:
- **Green:** Valid links
- **Red:** Broken links

Use Case: Great for a quick check on one page after you've made content edits.

Limitation: Only tests one page at a time, not the whole site.

Check My Links Demo Video

Practice Software Testing - Tool x +

practicesoftwaretesting.com

Practice Black Box Testing & Bug Hunting Testing Guide Bug Hunting

TOOLSHOP DEMO

Home Categories Contact Sign in EN



Sort

Price Range

Search

Search

Combination Pliers

Pliers

Bolt Cutters

CO₂: A B C D E

CO₂: A B C D E

CO₂: A B C D E

Check My Links Report

Practice Software Testing - Tool × CheckMyLinks_Report × +

Check My Links chrome-extension://aaajoalonednamcpodaocbfghdhcpbe/pages/links.html

Filter by Status Code: Show All ▾ Fetch All HTTPS Status Code Total Urls: 25

CSV Copy Excel Print Search:

URL	Status Code	Label	Action
https://github.com/testsmith-io/practice-software-testing...	200	GitHub repo	Copy Url View
https://practicesoftwaretesting.com/	200		Copy Url View
https://practicesoftwaretesting.com/	200	Home	Copy Url View
https://practicesoftwaretesting.com/#	200	Categories	Copy Url View
https://practicesoftwaretesting.com/#filters	200	Filters	Copy Url View
https://practicesoftwaretesting.com/auth/login	200	Sign in	Copy Url View
https://practicesoftwaretesting.com/category/hand...	200	Hand Tools	Copy Url View
https://practicesoftwaretesting.com/category/other	200	Other	Copy Url View
https://practicesoftwaretesting.com/category/power...	200	Power Tools	Copy Url View
https://practicesoftwaretesting.com/category/speci...	200	Special Tools	Copy Url View
https://practicesoftwaretesting.com/contact	200	Contact	Copy Url View
https://practicesoftwaretesting.com/privacy	200	Privacy Policy	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Combination Pliers A	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Pliers ABCDE\$12.01	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Bolt Cutters ABCDE\$4	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Long Nose Pliers ABC	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Slip Joint Pliers AB	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Claw Hammer with Shoe	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Hammer ABCDE\$12.58	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Claw Hammer ABCDE\$11	Copy Url View
https://practicesoftwaretesting.com/product/01KA7W...	200	Thor Hammer ABCDE\$11	Copy Url View

Approach 2: Advanced Audit ("Sitebulb")



What is it?

- A professional, powerful website crawler and SEO audit tool.
- It crawls your *entire* website, just like Google's bot would.
- (We used the 14-day full-featured trial).

What it provides:

- A complete dashboard and report of *all* broken links (internal and external) across the whole site.
- Provides much deeper analysis for a full, professional audit.

Sitebulb Demo Video

Trial License Sitebulb 9.1 Website Audit for <https://practicesoftwaretesting.com/>

Projects + New Project Compare Audits Single Page Analysis Structured Data Checker Help & Support Settings Your Account

Audit Overview URL Reports URL Explorer Link Explorer Site Visualisations PDF Report Bulk Exports Google Drive

All Hints (16) HTML Templates SEO Internal Links Indexability Redirects On Page Duplicate Content Response vs Render External XML Sitemap Generator

Audit Overview

Overview URLs

Audit Score 94 View Hints Triggered 16 **SEO Score** 96 View Hints Triggered 13

Crawled	View	Internal	View	External	View	Resources	Uncrawled
113		38		75		0	0

Audit Details

Date Nov 17, 2025 10:55 AM to Nov 17, 2025 10:57 AM

Start URL <https://practicesoftwaretesting.com/>

Project Toolshop

Maximum HTML Page Limit 500000

Maximum Crawl Depth 50

Crawl Type Chrome Crawler (Standard Audit)

User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/138.0.7204.101 Safari/537.36 (compatible; Sitebulb/1.1; +https://sitebulb.com)

Saved Export Files [View Folder](#)

[Start a New Audit](#) [Audit Settings](#) [Share Audit](#) [View Project](#) [Delete Audit](#)

Audit Notes

03

Usability & UX Testing

Usability and User Experience (UX)

The process of examining how effectively, efficiently, and satisfactorily a user can use a product to achieve specific goals, while also assessing the overall feeling and attitude of the user toward the system.

We check elements like ease of learning, intuitiveness, task completion time, error rates, and user satisfaction to ensure the product provides a positive and meaningful interaction.



UX vs. Usability

UX (User Experience) is the broad concept encompassing all aspects of the end-user's interaction with the company, its services, and its products, focusing on the overall feeling, attitude, and emotions.

Usability is a type of UX that specifically focuses on how easy it is to use an interface, measuring attributes like efficiency, effectiveness, and user satisfaction to achieve a specific goal.

Why is UX Testing Important?

Optimize User Flow: The primary goal is to find pain points and friction in the user journey, making the application efficient and satisfying to use.

Reduce Support Costs: By ensuring the product is intuitive and easy to learn, we minimize user confusion, which in turn significantly reduces the need for customer support and training.

Increase Adoption & Retention: We must verify that the user can easily achieve their goals, preventing frustration and increasing the likelihood of users returning and recommending the product.

Protect Investment & Revenue: A highly usable and enjoyable product directly contributes to higher conversion rates, ensuring the software's success and maximizing ROI.

How We Test: The UX Checklist

A **UX Checklist** is a predefined list of criteria used to validate the interface.

It helps testers systematically check elements like:

- Navigation
- Error Support & Feedback
- Clarity & Grouping
- Interaction Mechanics

The goal is to ensure that users can complete their tasks efficiently, without frustration, and have a positive experience throughout the entire process of using the software.

UX Checklist - User Experience

Usability informs the scenarios and tasks we design for our user surveys on platforms like Lookback.io.

UX Checklist - Navigation

Is there a link to the homepage on every page?

Does the user know where they are on the website?

Are all web pages/windows accessible from the menu?

Is the search function placed in the correct location?

Does the scroll bar appear when required?

Verify that all read-only fields are excluded from the tab order.

Verify that all disabled fields are excluded from the tab order.

Does the Tab order specified on the screen follow the order from top-left to bottom-right? (This is the default unless otherwise specified.)

When a form opens, is the first input field focused?

When an error message occurs, is the focus set on the first erroneous input field?

UX Checklist - Usability

Does the website clearly communicate its intended audience/purpose?

Does the website have a clear, recognizable "look and feel"?

Can regular users run the system without feeling frustrated?

Are all terms easily understood by the intended users of the page?

Does the system provide help or support to customers?

Is the font too large or too small to read?

Are the names on command buttons and option boxes abbreviated?

Are elements of the same type grouped together?

Are all lists sorted?

Is search suggestion/autocomplete supported?

Is tabbing supported?

Is there progress notification when processing long tasks (e.g., loading images, uploading files, loading pages)?

LOOKBACK.IO



Lookback.io is a powerful tool because it lets us capture and record real user interactions as they navigate a website or app, including their screen, facial expressions, and spoken thoughts.

For this seminar

We are focusing on using Lookback.io to test Usability, User Experience (UX), clarity of content, and the emotional response to the interface design.

DEMO

Goal: To comprehensively assess the User Experience (UX) of the application across different user roles (Guest, Normal, and Admin) and to identify specific areas for improvement in design and functionality.

Method

Guest and Normal User Assessment (Lookback.io Survey)

Admin User Assessment (One-on-One Qualitative Interview)

DEMO



Scenario (required)

Tell participants what their frame of mind should be before starting their first task.

You are trying to quickly find two items you need for a project. You need to use both the site's searching and filtering capabilities to complete this shopping trip efficiently.

Aa

Auto-open URL (optional)



practicesoftwaretesting.com/

DEMO

1

Task (required)



Task type

Task



Instructions

Begin by searching for the product type you need: a hammer.

Use the website/app's features (filters, sorting, etc.) to immediately narrow down the results to only show hammers that cost \$50.00 or less.

Review the options that meet your budget, inspect the details of at least one product, and then add your chosen hammer to your Shopping Cart.

Aa

Auto-open URL (optional)



example.com

DEMO

5

Task (required)



Task type

Single choice ▾

Question (single choice

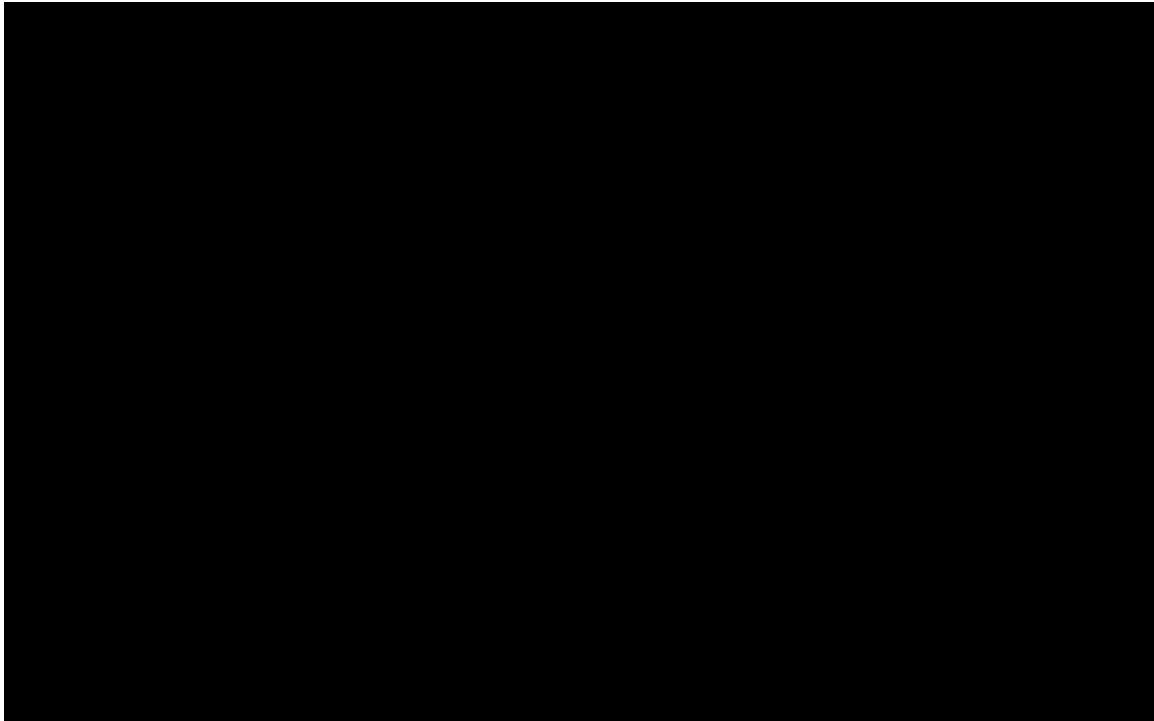
Finding the two items and applying the budget filter was an efficient process.

Aa

- Strongly Disagree
 - Disagree
 - Neutral
 - Agree
 - Strongly Agree

+ Add Option

Lookback Demo Video



Lookback Weaknesses

1. Session Limits

Free trial is restricted to only 5 sessions.

2. Limited Research Customization

Survey/question flows are less flexible compared to tools like Maze or UserTesting.

3. Limited Integrations

Few deep integrations with analytics or product management platforms.

Lacks advanced APIs for automation and workflow integration.

04

“AI-First” Approach

What is it?

Manual → Automation → Artificial Intelligence

AI performs recognition, validation, and decision-making – not predefined rules.

UI is treated like a human would: visually, semantically, behaviorally.



Traditional UI testing method

How tests identify elements: Uses fixed locators (XPath, CSS, IDs).

Stability: Very fragile — small UI changes break tests.

Validation method: Rule-based assertions (e.g., element exists).

Bug detection: Only detects failures explicitly coded.

Tester skill requirement: Must know scripting, locators, automation frameworks.

AI-First UI Testing testing method

How tests identify elements: Uses visual, semantic, and contextual understanding (like a human).

Stability: Highly stable — AI adapts to layout, text, or structural changes.

Validation method: Smart/visual assertions (layout, spacing, color, UX quality).

Bug detection: Detects visual bugs, UI drift, layout shifts, missing content.

Tester skill requirement: Can be done by QA + AI guidance; tech barrier is lower.

AI-First Approach: Mimic human action

Different from other types of testing—such as API or CI/CD tests—UI testing scenarios are far more **flexible** and must reflect how **real humans** perceive and interact with the application.

→ Not only to verify functionality, but to evaluate the overall user **experience, visuals, and context**. Modern AI models excel at this because they learn **visual patterns and contextual relationships** in a way that closely **mirrors human perception**, making the AI-first approach a **natural fit** for UI/UX testing.



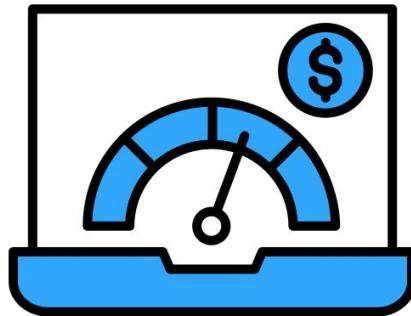
AI-First Approach: Utilize machine ability

Speed Advantage: Executes UI tests far faster than humans.

Expert Knowledge: Learns UI patterns, code logic, and human language.

Consistent Results: No fatigue, no variation, always reliable.

Multi-User Coverage: Adapts to different devices and user behaviors.



APPLITOOLS



 **applitools**

Applitools is an AI-powered visual testing platform designed to automatically **detect UI bugs** by comparing how an application should look versus how it actually looks.

For this seminar

We are focusing on using Applitools for **automated UI testing** with **Selenium** scripts. The platform's Visual AI analyzes each screen and detects regions with **abnormal appearance** compared to the established baseline. It **highlights** these areas and **identifies** potential UI errors caused by code changes, enabling us to quickly **spot visual regressions** and maintain a consistent user interface.

APPLITOOLS main function:

TOOLSHOP

Search Bar & Refining & Sort Buttons: [Refining Tools](#) [Sort Buttons](#)

Home | Log in | Contact | Sign in | 0 items

Searched for: Hammer

Sort: [Sort](#)

Price Range: [Price Range](#) [Search](#)

Filters: [Filters](#)

By category:

- Hand Tools
 - Claw hammer
 - Hammer
 - Wrecking
 - Shovel
 - Pliers
 - Wrenches
 - Drill
- Power Tools
 - Grinder
 - Drill
 - Screwdriver
 - Shears
 - Drill
- Other
 - Tool Belts
 - Storage Solutions
 - Workshop
 - Safety Gear
 - Measuring

By brand:

- Fingerline Tools
- MightyCraft Hammer

Sustainability:
 Show only eco-friendly products

This is a D940 application ([GitHub](#), [readme](#)), used for software testing/training purposes. | [Source this component](#) | [Privacy Policy](#) | [Release notes by Basem](#) ([issues on GitHub](#))

The screenshot shows a search results page for 'Hammer'. It includes a sidebar with sorting and filtering options, a main grid of hammer products with their names and prices, and a footer with copyright information.

TOOLSHOP

Search Bar & Refining & Sort Buttons: [Refining Tools](#) [Sort Buttons](#)

Home | Log in | Contact | Sign in | 0 items

Searched for: Hammer

Sort: [Sort](#)

Price Range: [Price Range](#) [Search](#)

Filters: [Filters](#)

By category:

- Hand Tools
 - Claw hammer
 - Hammer
 - Wrecking
 - Shovel
 - Pliers
 - Wrenches
 - Drill
- Power Tools
 - Grinder
 - Drill
 - Screwdriver
 - Shears
 - Drill
- Other
 - Tool Belts
 - Storage Solutions
 - Workshop
 - Safety Gear
 - Measuring

By brand:

- Thor Hammer
- Sledgehammer
- Claw Hammer with Fiberglass Handle
- Court Hammer

Sustainability:
 Show only eco-friendly products

This is a D940 application ([GitHub](#), [readme](#)), used for software testing/training purposes. | [Source this component](#) | [Privacy Policy](#) | [Release notes by Basem](#) ([issues on GitHub](#))

The screenshot shows the same search results for 'Hammer' as the first one, but with several UI elements highlighted in pink. These include the 'Sort' button, the 'Price Range' input, the 'Search' button, the 'Filters' button, the 'By category' section, the 'By brand' section, and the 'Sustainability' checkbox. The overall layout is identical to the first screenshot.

APPLITOOLS



 **applitools**

This application captures a screenshot during a test case and compares it with a baseline image

→ Significantly reducing the effort required to write scripts for validating every UI element in the browser.

Applitools Eyes supports many languages and frameworks, including Python, Java, JavaScript, C#, Ruby, Selenium, Playwright, Cypress, Appium, and more, and integrates with major CI tools, making it flexible for web, mobile, and desktop visual testing across different platforms.

Applitools Demo: Scenario

For this seminar

We conducted visual testing on the ToolShop text search feature by comparing images from two versions, **SPRINT5** and **SPRINT5_WITH-BUGS**. This approach clearly demonstrates the capability of Applitools' visual AI model to identify **differences between interface states**.

Applitools Demo: Video

The screenshot shows a web browser window with the URL eyes.applitools.com/app/test-results/?accountId=m55XvuM1z0mUNaX4qJ8wA_. The page title is "Test results". On the left, there is a sidebar with a "Last 0 batch runs" section and various icons. The main content area features a large red "APPLITOOLS DEMO" text. Below it is a cartoon illustration of two people, one at a desk with a computer and a potted plant, the other holding a tablet. A speech bubble above them says "Let's get started!". Below the illustration, the text "You don't have any test results yet" is displayed, followed by a green link: "Use the Learning Center > Getting Started section to learn the basics and create your first tests".



Thanks!