Cross Browser Testing

Performing Cross Browser Testing is essential for ensuring that applications look and function correctly across different browsers, devices, and operating systems. As a software testing consultancy firm, offering Cross Browser Testing as a core service helps us deliver a consistent, user-friendly experience to all end users—no matter how they access the app.

Here’s how our firm can strategically and efficiently perform cross browser testing to boost quality:

Objectives of Cross Browser Testing

Verify UI consistency and layout rendering

Ensure functionality works across different environments

Detect browser-specific bugs, rendering issues, or JavaScript errors

Test for responsive design on various screen sizes

Validate performance and load times across browsers

Step-by-Step Approach for Cross Browser Testing

1. Define the Scope

Identify target browsers based on client’s user analytics (e.g., Google Chrome 90+, Safari, Firefox, Edge, IE11 if needed).

Include mobile browsers (Safari on iOS, Chrome on Android).

Decide the depth of testing (e.g., full regression vs smoke test).

---

2. Prioritize Browsers and Devices

Use real user data (from Google Analytics or product telemetry) to determine:

Tier 1: Most-used browsers (must work flawlessly)

Tier 2: Less-used, but still important (basic functionality)

Tier 3: Legacy or niche browsers (limited testing or disclaimers)

---

3. Set Up the Testing Environment

Use a mix of:

Real devices for high-fidelity validation

Emulators/Simulators for quick checks

Cloud-based platforms for scalability

🔧 Popular tools:

BrowserStack

Sauce Labs

LambdaTest

CrossBrowserTesting.com

Selenium Grid (custom setup)

---

4. Automate Where Possible

Automate common test flows using:

Selenium WebDriver with browser capability configurations

TestCafe, Cypress (with plugins for cross-browser support)

Playwright (great for testing across Chromium, Firefox, WebKit)

💡 Use parallel test execution to save time and get faster feedback.

---

5. Manual Exploratory Testing

Perform manual visual and functional checks on real devices for:

UI rendering

Font rendering issues

Hover, scroll, and interactive element behaviors

Media playback (video/audio), file downloads

---

6. Check for Compatibility Issues

Focus on:

HTML/CSS rendering differences

JavaScript engine quirks

CSS grid/flexbox support

Font scaling, padding, margin differences

Form inputs and dropdown behaviors

---

7. Document and Report Issues Clearly

Provide screenshots or screen recordings of issues

Highlight browser-specific bugs with environment details

Suggest workarounds or CSS/JS fixes

🧾 **Sample report format:**

Browser	OS	Issue	Screenshot	Severity	Status
Safari 13	macOS	Button misaligned	Attached	Medium	Open

---

8. Integrate with CI/CD

Use CI/CD pipelines to run cross-browser tests automatically:

Trigger tests on pull requests or nightly builds

Fail builds if high-severity cross-browser issues are detected

🔧 Tools:

GitHub Actions + BrowserStack

Jenkins + Selenium Grid

GitLab CI + Sauce Labs

Value We Bring as a Testing Consultancy

Capability	Benefit for Client
Structured Test Matrices	Clear browser/device coverage
Tool Expertise	Save setup time, reduce costs
Automated + Manual Strategy	Balanced and reliable results

Capability	Benefit for Client
Visual Regression Testing	Detect pixel-level UI issues
Professional Reporting	Fast decision-making for devs and PMs

🛠 Want to Offer Premium Cross-Browser Testing?

We can offer:

Visual regression testing with Applitools or Percy

Accessibility testing (WCAG compliance across browsers)

Network throttling tests for low-bandwidth conditions

Localization checks across browsers and devices