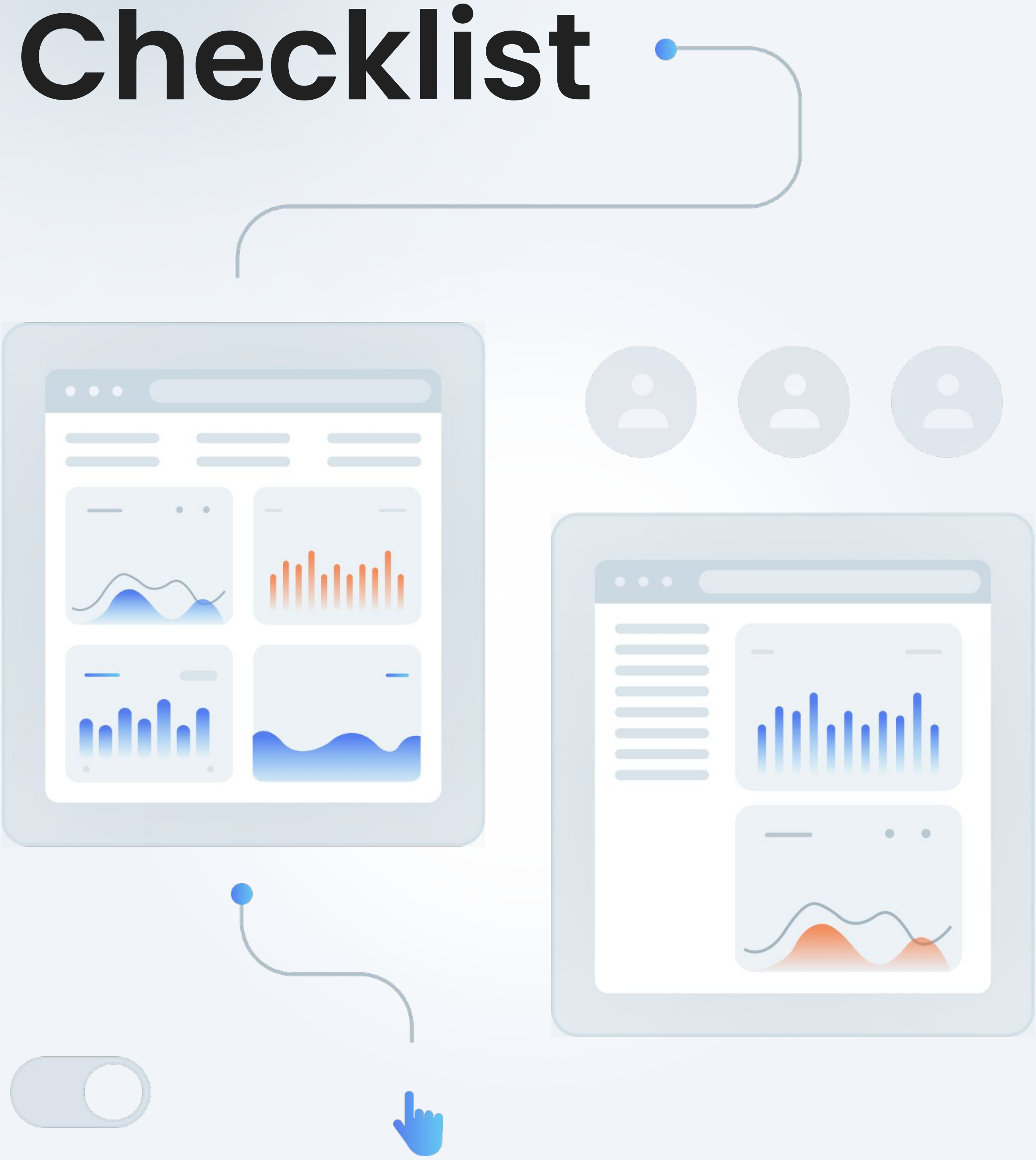
TESTFORT

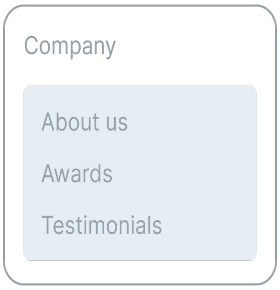
# UI/UX Testing Checklist



# UI/UX Testing Checklist

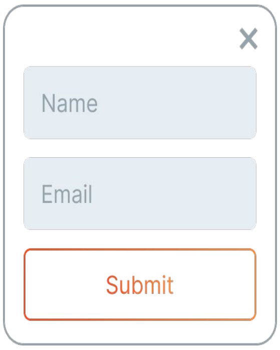
## 01 Functional Layer

### Navigation and links



- □ All navigation menus are working correctly
- □ Internal links lead to the correct pages
- □ External links open in new tabs/windows
- □ Back and forward browser buttons work as expected

### Forms and Data Input

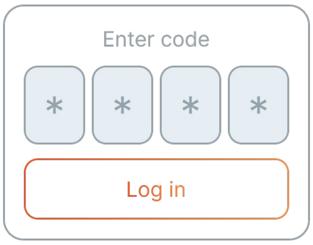


- □ All form fields accept appropriate input
- □ Required fields are properly validated
- □ Error messages are displayed for invalid inputs
- □ Forms submit successfully with valid data
- □ Auto-fill and auto-complete functions work correctly

### Search Functionality

- □ Search returns relevant results
- □ Search handles misspellings and partial matches
- □ Advanced search options (if applicable) function correctly

### User Actions



- □ User registration process works end-to-end
- □ Login/logout functionality is working
- □ Password reset process functions correctly
- □ User profile updates are saved and displayed accurately

**Charles Proxy**

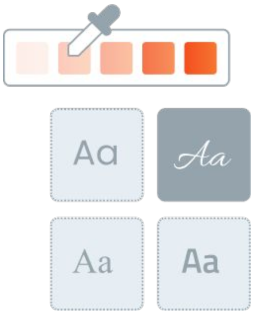
Use Charles Proxy to simulate slow networks and test your app's behavior under poor conditions.

**Prioritize testing**

your most common user journeys first. **88% of online consumers** are less likely to return to a site after a bad experience.

# 02 Visual Layer

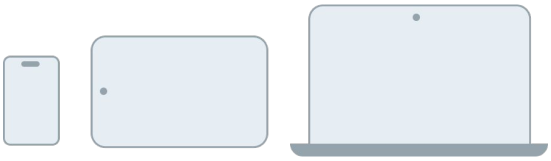
## Design Consistency



- □ Color scheme is consistent across all pages
- □ Typography (fonts, sizes, styles) is consistent
- □ Logos and branding elements are uniform
- □ Icons and buttons maintain a consistent style

## Responsive Design

- □ Layout adjusts appropriately for different screen sizes
- □ Images scale without distortion
- □ Text remains readable on all devices
- □ No horizontal scrolling on mobile devices



## Visual Hierarchy

- □ Important elements stand out visually
- □ Content is organized in a logical, easy-to-scan manner
- □ White space is used effectively to separate content

## Cross-Browser Compatibility

- □ Design renders correctly in all the browsers
- □ Functionality is consistent across all major browsers

**percy** **chromatic**

Use Percy or Chromatic for automated visual regression testing. They catch unintended visual changes across your entire UI.

**Often forgotten:**

Use Chrome's built-in print preview (Cmd + P) to test **how your UI looks when printed.** Critical for certain applications.

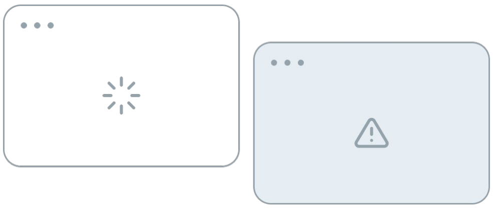
**LAMBDATEST**

LambdaTest allows you to perform live, interactive testing across 2000+ real browsers and operating systems.

# 03 Performance Layer

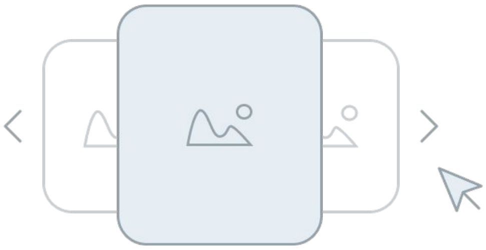
## Load Times

- □ Home page loads in under 3 seconds
- □ Other pages load in under 2 seconds
- □ Images and media are optimized for quick loading



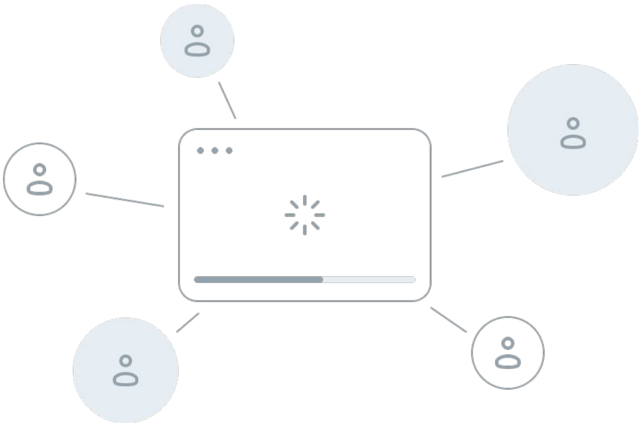
## Responsiveness

- □ Interactions (clicks, scrolls, hovers) respond immediately
- □ Animations and transitions are smooth
- □ No lag when typing in form fields



## Stability

- □ Application remains stable under heavy user load
- □ No memory leaks occur during prolonged use
- □ Application recovers well from interruptions



**A 1-second delay** in page load time can lead to a **7% loss in conversions.**

Optimize images and minify CSS/JavaScript for performance boost.

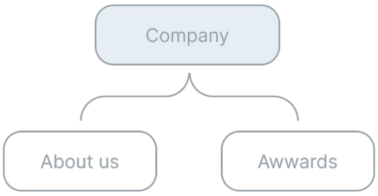
**Unoptimized fonts** can significantly slow down load times.

Use **Google's WebFont Loader** for better control.

# 04 User Experience Layer

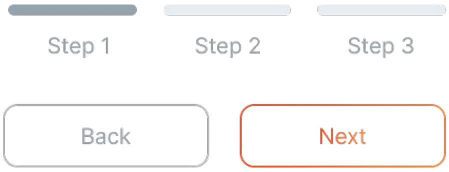
## Intuitiveness

- □ Users can accomplish key tasks without instruction
- □ Navigation structure is logical and easy to understand
- □ Important features are easily discoverable



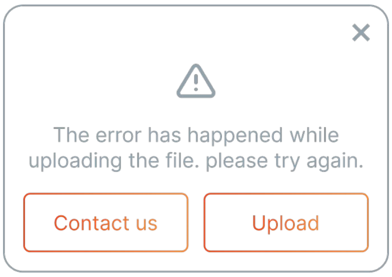
## Efficiency

- □ Common tasks can be completed in minimal steps
- □ Keyboard shortcuts are available for power users
- □ Auto-save features are implemented where appropriate



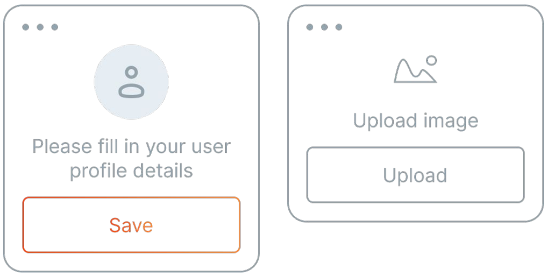
## Error Handling

- □ Error messages are clear and helpful
- □ Users can easily recover from errors
- □ Confirmation is required for irreversible actions


The error has happened while uploading the file. please try again.

## Content and Readability

- □ Content is concise and scannable
- □ Headings and subheadings are used effectively
- □ Technical jargon is minimized or explained


Please fill in your user profile details

crazyegg

**Cognitive load hack:**
Use tools like Crazy Egg to identify where users pause or abandon your site.

### Error messages

**Frame error messages positively.** 'Please enter a valid email' is better than 'Invalid email entered'.

### Squint test

Use the '**squint test**' — if you squint at your page, the most important elements should still stand out.

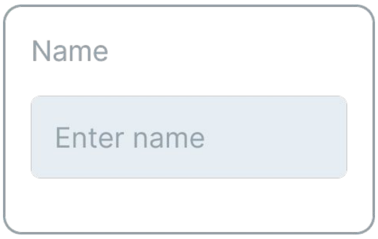
# 05 Accessibility Layer

## Screen Reader Compatibility

□ All content is accessible via screen readers

□ Images have appropriate alt text

□ ARIA labels are used where necessary

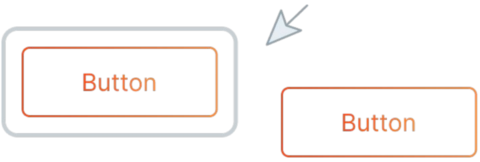


## Keyboard Navigation

□ All functionality is accessible via keyboard

□ Focus indicators are clearly visible

□ Logical tab order is maintained

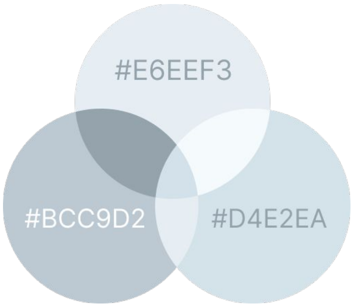


## Color and Contrast

□ Color is not the sole method of conveying information

□ Text has sufficient contrast with background colors

□ Interface is usable in high contrast mode

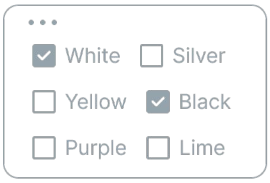


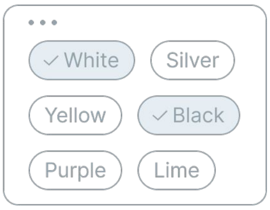
## Timing and Controls

□ Users can control time-sensitive content changes

□ Animations can be paused or disabled

□ No elements flash more than three times per second





The '**axe**' Chrome extension provides **instant accessibility feedback** as you develop.

**Accessibility:** from 'nice-to-have' to legal necessity.
**Pro tip:** Start with keyboard navigation — it's a quick win for accessibility.

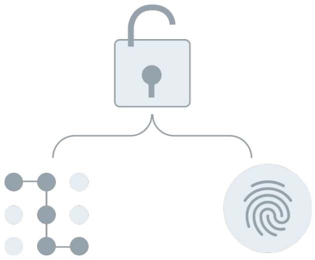
**Poor contrast** can reduce readability by up to 30%.

Use tools like **WebAIM's Contrast Checker** to ensure perceive your content easier.

# Additional Considerations

## Security

- □ All forms are protected against CSRF attacks
- □ Sensitive data is transmitted securely (HTTPS)
- □ User permissions are correctly implemented and enforced

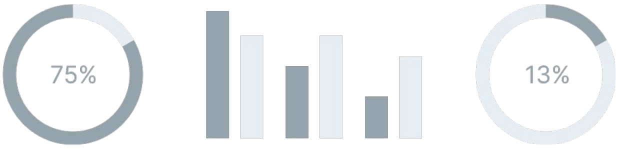


## Internationalization

- □ Application supports multiple languages (if applicable)
- □ Date, time, and number formats adjust to locale
- □ Text expands/contracts appropriately for different languages

## Performance Metrics

- □ Track and analyze user engagement metrics
- □ Monitor conversion rates for key user flows
- □ Regularly review and address user feedback



**Remember:** This checklist is a starting point. Tailor it to your specific product and user needs for best results.

# Common UI/UX Testing Challenges and Solutions

## Time Constraints

Challenge

Racing against deadlines, starting testing too late.

Solution

Implement continuous testing with Jenkins or GitLab CI in your CI/CD pipeline.

## Test Environment Setup

Challenge

Maintaining real-world environments for multiple devices.

Solution

Use cloud-based testing platforms like BrowserStack or Sauce Labs.

## Dynamic Content Issues

Challenge

Moving elements and unpredictable content loading breaking tests.

Solution

Use resilient locators (e.g., data-testid attributes) and tools like Cypress or Selenium WebDriver with explicit waits.

## Cross-Browser Inconsistencies

Challenge

UI looks different across browsers.

Solution

Utilize Chrome's device emulation, Responsinator, and cross-browser testing services.

## Performance Under Load

Challenge

UI slows down with increased user activity.

Solution

Automate performance tests with Lighthouse or WebPageTest. Simulate load using JMeter or Gatling.

## Visual Regression

Challenge

Unnoticed visual changes between builds.

Solution

Implement visual regression tools like Applitools or Percy for automated screenshot comparisons.

### Accessibility Oversights

| | |
|---|---|
| Challenge | Vague standards leading to missed accessibility issues. |
| Solution | Integrate axe-core or WAVE into your workflow. Use screen readers like NVDA or VoiceOver for manual checks. |

### Uncertain Usability

| | |
|---|---|
| Challenge | Difficulty predicting real user reactions. |
| Solution | Gather user feedback with UserTesting or Hotjar. Use heatmaps and session recordings for deeper insights. |

### UI Test Automation Complexity

| | |
|---|---|
| Challenge | Time-consuming automation that misses key issues. |
| Solution | Adopt a hybrid approach. Use Selenium for the web and Appium for mobile. Automate repetitive tasks, test complex flows manually. |

## Looking for custom actionable UI/UX test plan?

Contact us for free consultation!

Contact Us

Testfort is a QA and software development company with 23+ years of experience in the market. Our team doesn't just write code or run tests — we think like product owners. We offer flexible engagement models because we know one size doesn't fit all. Our team is focused on creating impact-driven solutions with your end goal in mind. We're not here to just tick boxes; we're here to make your product succeed.