Список використаних джерел

1. Емброуз Г. Основи. Графічний дизайн. Гевін Емброуз, Ніл Леонард /ArtHuss, 2019. – 534 p.
2. Графічний дизайн. Історія та сучасність. – Режим доступу: https://newmedia.ua/interesting-know-uk/hrafichnyy-dyzayn-istoriya-ta-suchasnist%CA%B9/
3. Штучний інтелект у графічному дизайні: як він змінює галузь? – Режим доступу: https://clickable.agency/ua/shtuchnij-intelekt-u-grafichnomu-dizajni/

AUTOMATION OF CROSS-BROWSER END-TO-END TESTING IN WEB DEVELOPMENT

ANASTASIIA TIUTIUN,
Course II, group 5, FIT SUTE, Ukraine
NATALIA KOTENKO,
Docent at the Department
of Software Engineering and Cybersecurity, SUTE
(https://orcid.org/0000-0002-2675-6514)

In the landscape of web development delivering a seamless user experience across different browsers and devices is always one of the highest priorities. Over time many types of testing have emerged that help front-end developers control the quality of their software. They are functional and compatibility testing strategies, that focus on compatibility and functional consistency across browsers, performance testing, accessibility control and regression testing.

However, one testing methodology stands out among others: end-to-end testing. While the aforementioned techniques play essential roles in verifying individual components and features, end-to-end testing provides a comprehensive evaluation of the entire application workflow from the user's point of view. This approach effectively simulates real-world scenarios and interactions across different browsers and environments.

Ensuring optimal website end-to-end testing across various Internet browsers used to be a time-consuming and labor-intensive process, often relying heavily on manual interactions. However, the agile development of automation tools and headless browsers has offered much more efficient and scalable solutions. Currently we can outline three ways of testing: simple scripting, headless browser testing and manual interaction.

Scripting involves creating special scripts that run on conventional browsers and simulate user actions by programmatically interacting with websites and their components, such as buttons or menus. While this is a good approach to automation, it is not as efficient as headless testing, which originated from it with the advent of headless browsers.

Headless testing involves running a browser application without the GUI («head»). Some browsers are created specifically to operate headless, while others can enter headless mode for testing and other purposes, such as web scraping. This mode allows the most widespread kinds of actions through command-line interface or network communication instead of usual visual interface. Headless browsers still parse HTML and CSS, as well as JavaScript and have full access to all sites on the web. Examples of headless browsers include PhantomJS, Chromium, Headless Chrome, Firefox Headless, and Edge Chromium.

Headless mode consumes fewer resources and offers better test performance by not loading any interface, while also providing better scalability. This strategy is perfect for network monitoring and testing regular user interactions such as filling out forms, mouse and keyboard events, navigation, network requests, interaction with APIs and browser events. However, headless browsers may not be the best choice for testing complex JavaScript flows that require the full browser application to function or are difficult to reproduce in test scripts. Even if certain long chains of actions can be introduced as a series of scripts, it will make it hard to maintain them and scale in the future.

Cross-browser testing, whether with or without headless mode, is achieved by setting up an environment where tests are run against multiple browser applications in parallel.

It is worth mentioning that manual testing is still widely applied because of limitations of automated end-to-end testing or in situations where time constraints prevent setting up a comprehensive testing environment. However, in the long run, a well-built environment gets much easier to maintain.

Overall, creating a system of tests is becoming easier as many companies develop specialized management applications for this purpose, and the integration of AI is making automated tests more accessible. For now, we could say that the best way to perform end-to-end cross-browser testing is a hybrid approach, where manual checks go alongside automation, helping shift development team's attention from more routine features to ones that require deeper control.

References

1. Lambdatest. Headless Browser Testing: Guide To 'What,' 'Why,' and 'How'. URL: https://www.lambdatest.com/learning-hub/headless-browser-testing (Last accessed: 10.04.2024)

2. Datadog Knowledge Center. Cross Browser Testing Overview. URL: https://www.datadoghq.com/knowledge-center/cross-browser-testing/ (Last accessed: 10.04.2024)

3. Automated storytelling technologies for cultural heritage / L. Colucci Cante et al. Advances in internet, data & web technologies. Cham, 2024. P. 597–606. URL: https://doi.org/10.1007/978-3-031-53555-0_57 (date of access: 14.04.2024).

4. Automated testing. Web-Teaching. Dordrecht. P. 235–241. URL: https://doi.org/10.1007/0-306-47682-7_15 (date of access: 14.04.2024).

5. Jaleel H. Q. Testing web applications. International journal of computer science and engineering. 2019. Vol. 6, no. 12. P. 1–9. URL: https://doi.org/10.14445/23488387/ijcse-v6i12p101 (date of access: 14.04.2024).

UI/UX DESIGN ТА FRONT-END DEVELOPMENT: РЕАЛІЇ СЬОГОДЕННЯ

ІГОР ЧЕРНЯЄВ,
студент 1курсу 5м групи,
Державний торговельно-економічний університет,
м. Київ, Україна
КАРИНА ХОРОЛЬСЬКА,
PhD, старший викладач кафедри інженерії
програмного забезпечення та кібербезпеки,
Державний торговельно-економічний університет,
м. Київ, Україна

У світі, де користувачі мають безліч варіантів для вибору веб-сайтів та додатків, конкуренція між компаніями за увагу та лояльність користувачів надзвичайно велика. Тому важливість ефективного UI/UX дизайну та Front-End Development стає критичною для створення привабливих та функціональних продуктів.

У сучасному світі важливо враховувати такі тенденції, як мінімалістичний дизайн, орієнтація на мобільні пристрої, персоналізація