

Spis treści

Przedmowa	11
O autorze	13
O korektorze	14
Wstęp	15

Część I. Frontendowe testowanie stron internetowych

Rozdział 1. Metody testowania z wykorzystaniem różnych przeglądarek	23
Przegląd krajobrazu internetowego	24
Typy aplikacji internetowych	25
Tradycyjne aplikacje internetowe	25
Responsywne aplikacje internetowe	26
Progresywne aplikacje internetowe	28
Typy testów dla aplikacji internetowych	30
Testy funkcjonalne aplikacji internetowych	31
Testy niefunkcjonalne aplikacji internetowych	31
Zastosowanie przeglądarek interfejsowych i bezinterfejsowych podczas tworzenia i testowania aplikacji	33
Wybór między przeglądarkami interfejsowymi i bezinterfejsowymi	34
Frameworki testowania przy użyciu przeglądarek bezinterfejsowych	36
Podsumowanie	39

Rozdział 2. Wyzwania stojące przed frontendowymi programistami aplikacji internetowych	40
Wyzwania związane z tworzeniem aplikacji internetowych	41
Jakość a cykl wydawniczy	43
Wyzwania związane z pokryciem dostępnych platform i systemów operacyjnych	43
Wyzwania niefunkcjonalne w tworzeniu aplikacji internetowych	48
Wyzwania związane z wydajnością	48
Wyzwania związane z ułatwieniami dostępu	49
Wyzwania związane z zapewnianiem zgodności aplikacji internetowych	51
Podsumowanie	53
Rozdział 3. Najlepsze frameworki do automatyzacji testów aplikacji internetowych	54
Przegląd rynku testowania aplikacji internetowych	55
Pierwsze kroki z frameworkiem Selenium WebDriver	56
Konfigurowanie komponentu WebDriver	57
Selenium Grid	60
Pierwsze kroki z frameworkiem Cypress	63
Ważne funkcjonalności Cypressa	65
Pierwsze kroki z frameworkiem Google Puppeteer	68
Pierwsze kroki z frameworkiem Microsoft Playwright	69
Podsumowanie	72
Rozdział 4. Dopasowanie odpowiednich osób i przypadków użycia do frameworków testowych	73
Wymagania techniczne	74
Przegląd osób testujących strony internetowe	74
Przypadki użycia i zagadnienia dotyczące wyboru solidnego frameworku do automatyzacji testów	76
Kwestie społecznościowe	76
Możliwości testowania na dużą skalę	77
Integracje stosu narzędzi i wtyczki	77
Łatwość użytkowania i popularność	77
Wielokrotny użytek i łatwość utrzymywania	78
Raportowanie, analiza testów i sztuczna inteligencja	78
Macierz oceny frameworków testowych	79
Podsumowanie	83
Rozdział 5. Wprowadzenie do wiodących frontendowych frameworków do tworzenia aplikacji internetowych	85
Wymagania techniczne	86
Wprowadzenie do wiodących frameworków do tworzenia aplikacji internetowych	86
Wtyczne dotyczące wyboru frameworku do tworzenia aplikacji internetowych	87
ReactJS	88
AngularJS	91
Vue.js	93

Ember.js	95
Svelte	97
Podsumowanie	99

Część II. Strategia ciągłego testowania dla programistów aplikacji internetowych

Rozdział 6. Filary strategii programistycznych testów aplikacji internetowych 103

Filary planu i strategii testowania aplikacji internetowej	104
Poznaj użytkowników docelowych	105
Opracuj plan testów	105
Przygotuj stos narzędzi i środowiska	106
Ustal kryteria i cele jakości	106
Określ porządek chronologiczny i harmonogram	106
Wykonuj, monitoruj, mierz i dokumentuj	106
Pomiar sukcesu strategii ciągłego testowania	107
Studium przypadku — strategia testowania rzeczywistej aplikacji internetowej	110
Podsumowanie	111

Rozdział 7. Podstawowe funkcjonalności wiodących frameworków do automatyzacji testów javascriptowych 112

Porównanie funkcjonalności frameworków do automatyzacji testów	113
Testy wizualne	113
Testowanie interfejsów API	116
Obsługiwane języki programowania	118
Testowanie urządzeń mobilnych	119
Testy wydajnościowe	121
Testowanie ułatwień dostępu	124
Testowanie żądań sieciowych i atrapy usług	125
Praca z elementami	127
Istotne zdarzenia wymagające ponownej ewaluacji frameworków do automatyzacji testów	133
Podsumowanie	134

Rozdział 8. Mierzenie pokrycia testowego aplikacji internetowej 135

Wprowadzenie do pokrycia kodu i pokrycia testowego	136
Pokrycie testowe	136
Pokrycie kodu	137
Narzędzia mierzenia pokrycia kodu JavaScriptu dla programistów aplikacji internetowych	138
Pomiar pokrycia kodu JavaScriptu za pomocą narzędzia Istanbul i frameworku Cypress	139
Uzupełnianie pokrycia kodu pokryciem testowym	142
Podsumowanie	143

Część III. Przewodnik po frontendowych frameworkach do automatyzacji javascriptowych testów aplikacji internetowych

Rozdział 9. Praca z frameworkiem Selenium	147
Wymagania techniczne	147
Framework Selenium i jego komponenty	148
Selenium WebDriver	148
Zaawansowane funkcjonalności Selenium	151
Różne metody testowania z wykorzystaniem Selenium	155
Przyszłość frameworku Selenium	159
Podsumowanie	160
Rozdział 10. Praca z frameworkiem Cypress	161
Wymagania techniczne	162
Pierwsze kroki z Cypressem	162
GUI frameworku Cypress	162
IDE i wiersz poleceń Cypressa	164
Zaawansowane funkcjonalności automatyzacji testów Cypressa	167
Ponawianie testów Cypressa	167
Korzystanie z namiastek, szpiegów i zegarów w Cypressie	168
Uruchamianie Cypressa w ramach potoku CI	170
Testowanie komponentów	171
Cypress Studio	172
Wtyczki Cypressa	175
Testowanie API z wykorzystaniem Cypressa	176
Przyszłość frameworku Cypress	178
Podsumowanie	179
Rozdział 11. Praca z frameworkiem Playwright	180
Wymagania techniczne	181
Pierwsze kroki z Playwrightem	181
Zaawansowane funkcjonalności automatyzacji testów Playwrighta	184
Narzędzie Inspector Playwrighta	184
Emulowanie urządzeń mobilnych	186
Adnotacje testowe Playwrighta	186
Testowanie API przy użyciu Playwrighta	188
Asercje Playwrighta	189
Atrapy żądań sieciowych Playwrighta	190
Obiektowy Model Strony Playwrighta	191
Raporty testowe Playwrighta	191
Test runnery Playwrighta	192
Trace viewer Playwrighta	193
Zaawansowane konfiguracje Playwrighta	194
Integracja Playwrighta z CI	196
Przyszłość frameworku Playwright	197
Podsumowanie	199

Rozdział 12. Praca z frameworkiem Puppeteer	200
Wymagania techniczne	201
Pierwsze kroki z Puppeteerem	201
Zaawansowane funkcjonalności automatyzacji testów Puppeteera	205
Przestrzenie nazw Puppeteera	205
Praca z elementami przy użyciu Puppeteera	206
Testy obciążeniowe Puppeteera	209
Puppeteer i Cucumber BDD	209
Testy ułatwień dostępu Puppeteera	210
Śledzenie aplikacji internetowych przy użyciu Puppeteera	211
Testy API Puppeteera	212
Puppeteer i narzędzia dla programistów Google'a	212
Integracja Puppeteera z frameworkiem CodeceptJS	213
Integracja Puppeteera z CI	215
Przyszłość frameworku Puppeteer	217
Podsumowanie	219
Rozdział 13. Uzupełnianie testów opartych na kodzie automatyzacją niskokodową	220
Podstawowe funkcjonalności narzędzi do testowania niskokodowego i bezkodowego	221
Przegląd narzędzi bezkodowych w krajobrazie open source	224
Narzędzia bezkodowe na licencji open source	224
Wiodące komercyjne narzędzia do bezkodowego testowania aplikacji internetowych	230
Narzędzie Perfecto Scriptless Web	231
Narzędzie Testim do bezkodowego testowania aplikacji internetowych	234
Narzędzie Mabl do bezkodowego testowania aplikacji internetowych	237
Podsumowanie	242
Rozdział 14. Podsumowanie	244
Główne wnioski z książki	245
Przydatne materiały referencyjne	246
Dla frameworku Cypress	246
Dla frameworku Playwright	247
Dla frameworku Selenium	247
Dla frameworku Puppeteer	247