

Spis treści

Wstęp	7
1. Tworzenie aplikacji	11
1.1. Generowanie prostej aplikacji	11
1.2. Tworzenie aplikacji o bogatych treściach z zastosowaniem narzędzia Gatsby	15
1.3. Tworzenie uniwersalnych aplikacji przy użyciu Razzle	19
1.4. Zarządzanie kodem klienta i serwera z wykorzystaniem Next.js	21
1.5. Tworzenie małych aplikacji przy użyciu Preacta	23
1.6. Tworzenie bibliotek z wykorzystaniem nwb	27
1.7. Dodawanie Reacta do aplikacji Rails za pomocą Webpackera	29
1.8. Tworzenie niestandardowych elementów przy użyciu Preacta	31
1.9. Tworzenie komponentów z zastosowaniem Storybooka	35
1.10. Testowanie kodu w przeglądarce z zastosowaniem Cypressa	38
2. Routing	41
2.1. Tworzenie interfejsu z responsywnymi trasami	41
2.2. Przenoszenie stanu do tras	48
2.3. Użycie MemoryRouter do wykonywania testów jednostkowych	53
2.4. Stosowanie potwierżeń opuszczenia strony	56
2.5. Tworzenie przejść przy użyciu biblioteki React Transition Group	62
2.6. Tworzenie zabezpieczonych tras	67
3. Zarządzanie stanem	73
3.1. Stosowanie reduktorów do zarządzania złożonym stanem	73
3.2. Implementacja mechanizmu cofania	81
3.3. Tworzenie i walidacja formularzy	88
3.4. Pomiar upływu czasu przy użyciu zegara	95
3.5. Monitor stanu połączenia z internetem	99
3.6. Zarządzanie globalnym stanem przy użyciu Reduxa	101
3.7. Zachowywanie danych podczas przeładowywania stron przy użyciu Redux Persist	107
3.8. Obliczanie stanu pochodnego przy użyciu Reselect	111
4. Projektowanie interakcji	116
4.1. Tworzenie scentralizowanej obsługi błędów	116
4.2. Tworzenie interaktywnego przewodnika	121
4.3. Zastosowanie reduktorów do tworzenia złożonych interakcji	128
4.4. Dodawanie interakcji korzystających z klawiatury	133
4.5. Stosowanie formatu Markdown do tworzenia bogatych treści	137
4.6. Animacje tworzone z użyciem klas CSS	142

4.7. Tworzenie animacji z użyciem React Animation	144
4.8. Tworzenie infografik przy użyciu biblioteki TweenOne	149
5. Połączenia z usługami	155
5.1. Przekształcenie wywołania sieciowego w hook	155
5.2. Automatyczne odświeżanie przy użyciu liczników stanu	161
5.3. Anulowanie żądań sieciowych z wykorzystaniem tokenów	169
5.4. Generowanie żądań sieciowych z użyciem oprogramowania pośredniego Reduxa	172
5.5. Nawiązywanie połączenia z GraphQL	178
5.6. Ograniczanie obciążenia sieci poprzez opóźnianie generowania żądań	185
6. Biblioteki komponentów	189
6.1. Stosowanie Material Design przy użyciu biblioteki Material-UI	190
6.2. Tworzenie prostego interfejsu użytkownika za pomocą React Bootstrap	197
6.3. Przeglądanie zbiorów danych przy użyciu React Window	201
6.4. Tworzenie responsywnych okien dialogowych z wykorzystaniem Material-UI	203
6.5. Tworzenie konsoli administracyjnej przy użyciu React Admin	206
6.6. Nie masz projektanta? Użyj Semantic UI	213
7. Bezpieczeństwo	219
7.1. Zabezpieczaj żądania, nie trasy	219
7.2. Uwierzytelnianie z użyciem fizycznych tokenów	228
7.3. Włączanie protokołu HTTPS	238
7.4. Uwierzytelnianie za pomocą odcisków palców	241
7.5. Stosowanie logowania potwierdzającego	247
7.6. Stosowanie uwierzytelniania jednoskładnikowego	254
7.7. Testowanie na urządzeniu z Androidem	259
7.8. Sprawdzanie bezpieczeństwa/zabezpieczeń przy wykorzystaniu ESLint	261
7.9. Dostosowywanie formularzy logowania pod kątem przeglądarek	265
8. Testowanie	268
8.1. Stosowanie React Testing Library	269
8.2. Stosowanie Storybooka do testów renderowania	276
8.3. Testowanie bez serwera z użyciem Cypressa	282
8.4. Stosowanie Cypressa do testowania aplikacji bez połączenia z internetem	289
8.5. Testowanie w przeglądarce przy użyciu Selenium	292
8.6. Testowanie prezentacji w różnych przeglądarkach z użyciem ImageMagick	299
8.7. Dodawanie konsoli do przeglądarek mobilnych	307
8.8. Usuwanie losowości z testów	311
8.9. Podróż w czasie	314

9. Dostępność	320
9.1. Stosowanie punktów orientacyjnych	321
9.2. Stosowanie ról oraz atrybutów alt i title	326
9.3. Sprawdzanie dostępności za pomocą narzędzia ESLint	334
9.4. Stosowanie axe DevTools podczas działania aplikacji	340
9.5. Automatyzowanie testów w przeglądarce za pomocą narzędzia Cypress Axe	345
9.6. Dodawanie przycisków pomijania	349
9.7. Dodawanie pomijania obszarów strony	355
9.8. Przechwytywanie zasięgu w modalnym oknie dialogowym	362
9.9. Tworzenie czytnika ekranu za pomocą Speech API	365
10. Wydajność	370
10.1. Stosowanie narzędzi mierzenia wydajności działających w przeglądarce	371
10.2. Śledzenie renderowania za pomocą Profilera	378
10.3. Tworzenie testów jednostkowych z użyciem komponentu Profiler	383
10.4. Precyzyjny pomiar czasu	388
10.5. Zmniejszanie aplikacji z wykorzystaniem dzielenia kodu	391
10.6. Łączenie obietnic żądań sieciowych	398
10.7. Stosowanie renderowania po stronie serwera	401
10.8. Stosowanie metryk web vitals	411
11. Progresywne aplikacje internetowe	414
11.1. Stosowanie mechanizmu service workers przy użyciu narzędzia Workbox	415
11.2. Tworzenie PWS za pomocą narzędzia create-react-app	430
11.3. Przechowywanie zasobów zewnętrznych w pamięci podręcznej	433
11.4. Automatyzacja odświeżania skryptów service worker	437
11.5. Dodawanie powiadomień	442
11.6. Wprowadzanie zmian offline z zastosowaniem synchronizacji w tle	449
11.7. Dodawanie niestandardowego instalacyjnego interfejsu użytkownika	454
11.8. Dostarczanie odpowiedzi w trybie offline	458