Plan testów dla test.testowanie-oprogramowanie.pl

Spis treści:

1. Wstęp
2. Zakres testów
3. Przedmiot testów
4. Kryteria zaliczenia
5. Kryteria niezaliczenia
6. Kryteria wejścia
7. Kryteria wyjścia
8. Lista funkcjonalności do przetestowania
9. Środowisko testowe
10. Kategorie błędów
11. Miejsce testów
12. Harmonogram testów
13. Raporty z testów
14. Lista narzędzi
15. Zarządzanie incydentami
16. Role i odpowiedzialność

**Wstęp**

Celem tworzonego planu testów jest szczegółowe przetestowanie funkcjonalności sklepu w oparciu o dostarczone wymagania.

Adres sklepu to <https://test.testowanie-oprogramowania.pl/>

Wszystkie napotkane defekty zostaną zgłoszone, poprawione i przekazane testom do testów potwierdzających.

Dodatkowym celem tworzonego testów jest nauka i praktyka testowania.

**Zakres testów**

1. Testowanie funkcjonalności:
   1. Rejestracja nowego konta użytkownika
   2. Logowanie się na istniejące konto
   3. Przypomnienie hasła/resetowanie hasła
2. Testowanie interfejsu użytkownika:
   1. Sprawdzenie czy interfejs jest intuicyjny i łatwy w nawigacji
   2. Upewnienie się, że wszystkie elementy interfejsu (przyciski, linki, pola tekstowe) działają poprawnie
   3. Testowanie responsywności strony na różnych urządzeniach (komputer, tablet, telefon)
3. Testowanie zgodności:
   1. Upewnienie się, że strona działa poprawnie na różnych przeglądarkach internetowych (Chrome, Firefox, Safari, Opera GX itp.).
   2. Sprawdzenie zgodności z różnymi systemami operacyjnymi (Windows, macOs, Linux, iOS, Android).
4. Testowanie wydajności:
   1. Pomiar czasu ładowania się strony
   2. Testowanie responsywności strony pod obciążeniem (np. duże obrazy, duże ilości danych, zwiększona liczba użytkowników)
5. Testowanie bezpieczeństwa:
   1. Sprawdzenie czy strona jest odporna na podstawowe ataki
   2. Weryfikacja poprawności usług sesji i uprawnień dostępu
6. Testowanie kompatybilności:
   1. Upewnienie się, że strona jest kompatybilna z różnymi wersjami języka HTML, CSS, JavaScript
   2. Testowanie zgodności z różnymi wersjami serwerów WWW (Apache itp.)
7. Testowanie zawartości:
   1. Sprawdzenie poprawności gramatycznej i stylistycznej tekstu na stronie
   2. Weryfikacja poprawności linków
   3. Upewnienie się, że obrazy i multimedia są wyświetlane poprawnie

**Przedmiot testów**

Przedmiotem testów jest sklep internetowy dostępny pod adresem <https://test.testowanie-oprogramowania.pl/> wraz z jego wszystkimi funkcjonalnościami.

**Kryterium zaliczenia**

1. Stabilność:
   1. Strona musi być stabilna i nie powinna wykazywać poważnych błędów, które uniemożliwiają użytkownikowi korzystanie z jej funkcji.
2. Zgodność z wymaganiami:
   1. Strona powinna spełniać wszystkie określone w specyfikacji wymagania funkcjonalne i niefunkcjonalne
3. Wygląd i interfejs użytkownika:
   1. Interfejs użytkownika powinien być atrakcyjny wizualnie i intuicyjny w obsłudze
   2. Strona powinna być responsywna i dobrze wyglądać na różnych urządzeniach i przeglądarkach
4. Wydajność:
   1. Strona powinna ładować się szybko nawet przy dużej liczbie użytkowników 500 ms
   2. Czas odpowiedzi na interakcję użytkownika powinien być nie dłuższy niż 500 ms
   3. Strona powinna wytrzymać obciążenia 500 osób w tym samym czasie
5. Bezpieczeństwo:
   1. Strona powinna być odporna na ataki typu SQL Injection, CSRF
   2. Dane użytkowników powinny być bezpieczne i dobrze zabezpieczone
6. Dokumentacja:
   1. Istniejąca dokumentacja powinna być aktualna i dokładnie odzwierciedlać funkcjonalności strony
   2. Wszelkie zmiany wprowadzone w trakcie procesu testowania powinny być odpowiednio udokumentowane
7. Zgodność z standardami:
   1. Strona powinna być zgodna z obowiązującymi standardami internetowymi takimi jak HTML5, CSS3
8. Zgodność z planem testów:
   1. Wszystkie przewidziane testy powinny być przeprowadzone, a wyniki powinny być zgodne z oczekiwaniami.
9. Zgodność z harmonogramem:
   1. Testowanie strony powinno być zakończone w wyznaczonym czasie, zgodnie z ustalonym harmonogramem

**Kryterium niezaliczenia**

1. Błędy krytyczne
   1. Istnienie krytycznych błędów uniemożliwiających podstawowe funkcje strony, takich jak rejestracja, logowanie, przeglądanie zawartości itp.
2. Brak zgodności z wymaganiami
   1. Strona nie spełnia kluczowych wymagań funkcjonalnych lub niefunkcjonalnych określonych w specyfikacji
3. Powtarzające się błędy:
   1. Występowanie wielu powtarzających się błędów
4. Naruszenie zasad bezpieczeństwa:
   1. Istnienie poważnych luk w zabezpieczeniach, które mogą narażać dane użytkownika na ryzyko kradzieży lub utraty
5. Wydajność poniżej akceptowanego poziomu:
   1. Strona ma zbyt długi czas ładowania się lub reakcji na interakcję użytkownika (> 500 ms)
6. Błędy związane z interfejsem użytkownika:
   1. Interfejs użytkownika jest nieintuicyjny, trudny w obsłudze lub niezgodny z oczekiwaniami użytkownika
7. Brak zgodności z przeglądarkami i urządzeniami:
   1. Strona nie działa poprawnie na jednej lub więcej przeglądarek internetowych lub urządzeniach
8. Błędy związane z treścią:
   1. Treść strony zawiera błędy gramatyczne, ortograficzne lub merytoryczne
9. Błędy zgodności z standardami:
   1. Strona nie spełnia obowiązujących standardów internetowych, takich jak HTML5, CSS3
10. Brak dokumentacji lub dokumentacja nieaktualna:
    1. Brak aktualnej dokumentacji strony lub dokumentacja nie odzwierciedla rzeczywistych funkcjonalności i zachowań strony

**Kryterium wejścia**

* Kompletna dokumentacja produktu, specyfikacje funkcjonalne, instrukcje obsługi itp.
* Wszystkie funkcje zaimplementowane i gotowe do testowania
* Środowiska testowe skonfigurowane i dostępne dla zespołu testowego
* Środowisko testowe musi być zbliżone do środowiska produkcyjnego
* Dostępność danych testowych
* Dostęp do urządzenia z systemem Windows, macOs, Linux, iOS, Android
* Dostęp do przeglądarek internetowych Chrome, Firefox, Safari, Opera GX, Edge
* Zgoda na rozpoczęcie testów
* Zdefiniowane kryterium zaliczenia i niezaliczenia

**Kryterium wyjścia**

* Wszystkie kryteria zaliczenia określone w planie testów muszą zostać spełnione. Strona musi działa zgodnie z oczekiwaniami klienta i spełnia wszystkie funkcjonalne i niefunkcjonalne wymagania
* Brak krytycznych błędów
* Strona powinna być wydajna i responsywna, ładować się szybko i sprawnie reagować na interakcję użytkownika
* Wszystkie potencjalne luki bezpieczeństwa powinny zostać załatane, a strona powinna być odporna na typowe ataki
* Zgodność z przeglądarkami i urządzeniami
* Przygotowana dokumentacja dla zespołu deweloperskiego i wsparcia technicznego
* Zgoda na wydanie
* Wszystkie zgłoszone błędy zostały zarejestrowane i przeanalizowane.
* Zgodność z harmonogramem i budżetem
* Ocena ryzyka

**Lista funkcjonalności do przetestowania**

* Rejestracja użytkownika
* Logowanie użytkownika
* Przeglądanie zawartości
* Resetowanie hasła
* Funkcjonowanie koszyka (dodawanie, usuwanie, aktualizowanie produktu w koszyku)
* Płatność za produkt z koszyka
* Kompatybilność aplikacji z różnymi przeglądarkami oraz urządzeniami mobilnymi
* Formularz kontaktowy

**Środowisko testowe**

**Środowisko testowe**

Testy sklepu internetowe będą przeprowadzane na dedykowanym serwerze <https://skleptest.pl/>

Środowisko testowe jest zbliżone parametrami do środowiska produkcyjnego.

Specyfikacja:

Procesor: 2 GHz

Ram: 4 GB

Dysk: 100 GB

System: Windows 11 Home 64-bitowy

Testy będą przeprowadzane przy użyciu poniższych urządzeń:

1. Laptop Huawei MRC-WX0:
   1. System: Windows 10 Home 64-bitowy
   2. Procesor: Intel Core i5 1.6-3.4GHz
   3. Ram: 16 GB
   4. Karta graficzna: NVIDIA GeForce MX150
   5. Dysk: 104 GB
2. Urządzenie mobilne Oppo Reno 3 Pro:
   1. System: Android 10
   2. Procesor: Qualcomm Snapdragon 765G
   3. Ram: 12 GB
   4. Rozdzielczość ekranu: 2400 x 1080 pikseli
3. Tablet Lenovo Tab M10 Plus 3rd Gen:
   1. System Android 11
   2. Procesor: UNISOC Spreadtrum T610 (2x 1.8 GHz, A75 + 6x 1.8 GHz, A55)
   3. Ram: 4 GB
   4. Rozdzielczość ekranu: 1920 x 1200

**Kategorie błędów**

* Błędy funkcyjne
* Błędy interfejsu użytkownika
* Błędy wydajnościowe
* Błędy bezpieczeństwa
* Błędy kompatybilności
* Błędy treści

**Miejsce testów**

Testy będą przeprowadzane w siedzibie firmy wytwarzającej oprogramowanie przez testera

**Harmonogram testów**

1. Faza przygotowawcza (Czas 24h):
   1. przygotowanie środowiska testowego, w tym instalacja niezbędnych narzędzi i skonfigurowanie danych testowych
   2. Przygotowanie planu testów, w tym sprecyzowanie przypadków testowych dla różnych funkcjonalności strony
   3. Przygotowanie dokumentacji testowej, w tym kryteriów zaliczenia i niezaliczenia oraz kategorii błędów
2. Faza testowania (Czas 60h):
   1. Testowanie funkcjonalności strony, zgodnie z przygotowanymi przypadkami testowymi
   2. Rejestracja i raportowanie znalezionych błędów, w tym ich kategorii, priorytetu i opisu
   3. Testowanie wydajności i responsywności strony na różnych urządzeniach i przeglądarkach
   4. Testowanie zgodności z różnymi standardami i przepisami
3. Faza retestów i napraw (Czas 50h):
   1. Ponowne przetestowanie funkcjonalności strony, które zostały uznane za błędne w poprzedniej fazie testów
   2. Weryfikacja naprawionych błędów i sprawdzenie czy zostały one skorygowane zgodnie z oczekiwaniami
   3. Testowanie integracji naprawionych funkcjonalności z istniejącymi
4. Faza akceptacji (Czas 15h):
   1. Przeprowadzenie testów akceptacyjnych z udziałem klienta lub osób odpowiedzialnych za projekt
   2. Ocena wyników testów i zatwierdzenie gotowości strony do wydania
5. Faza dokumentacji (Czas 15h):
   1. Aktualizacja dokumentacji testowej
   2. Przygotowanie dokumentacji, instrukcji obsługi i innych niezbędnych dokumentów
6. Faza wydania (Czas 15h):
   1. Wydanie strony internetowej po zatwierdzeniu przez klienta
   2. Monitorowanie działania strony po wydaniu i reakcja na ewentualne problemy zgłoszone przez użytkowników

Przybliżony czas to 179 godzin, który może się zmienić w zależności od stopnia realizacji testów.

**Raporty z testów**

* Raport z testów funkcjonalnych
* Raport z testów bezpieczeństwa
* Raport z testów kompatybilności
* Raporty o wykrytych defektach
* Zaprojektowane przypadki testowe
* Skrypty testów automatycznych
* Metryki z testów wydajnościowych

**Lista narzędzi**

1. Narzędzie do raportowania defektów:
   1. Jira
2. Narzędzia do tworzenia przypadków testowych:
   1. TestLink
   2. XRay
3. Narzędzia do testowania funkcjonalności:
   1. Cypress
   2. Selenium WebDriver
4. Narzędzie do testowania wydajności:
   1. Apache JMeter
5. Narzędzia do testowania bezpieczeństwa:
   1. Burp Suite
   2. OWASP ZAP
6. Narzędzia do testowania kompatybilności:
   1. BrowserStack
   2. LampdaTest
7. Narzędzia do automatyzacji testów:
   1. Selenium WebDriver
   2. Appium
8. Narzędzie do monitorowania wydajności:
   1. Grafana

**Zarządzanie incydentami**

Po wykryciu defektu tester taki przypadek zgłasza w programie Jira. Będzie ustalony priorytet oraz osobę, która będzie zajmować się poprawieniem takiego błędu.

Po poprawieniu takiego błędu programista przekaże testerowi stosowną informację do przeprowadzenia testów potwierdzających.

W momencie poprawnego naprawienia defektu, zgłoszenie zostanie oznaczone jako zamknięte i przeniesione do kolumny “Gotowe”.

W przypadku, gdy defekt ponownie się pojawi z powodu niepoprawnego naprawienia lub z innego podobnego powodu, zgłoszenie zostanie na nowo zostanie przekazane programiście.

**Role i odpowiedzialność**

1. Kierownik projektu:
   1. Koordynuje cały proces testowania.
   2. Zarządza zasobami, harmonogramem i budżetem testów.
   3. Monitoruje postępy i raportuje o wynikach testów.
2. Lider zespołu testowego:
   1. Zarządza zespołem testowym i koordynuje jego działania.
   2. Planuje strategię testowania i określa priorytety.
   3. Nadzoruje przygotowanie dokumentacji testowej.
3. Analityk testów:
   1. Analizuje wymagania klienta i specyfikację projektu.
   2. Przygotowuje przypadki testowe na podstawie ustalonych kryteriów akceptacji.
   3. Koordynuje zespołem testerów i programistów w celu zrozumienia funkcjonalności do przetestowania.
4. Tester funkcjonalny:
   1. Przeprowadza testy funkcjonalne, zgodnie z ustalonym planem testów.
   2. Rejestracja znalezionych błędów i ich raportowanie.
   3. Wykonywanie retestów po naprawie zgłoszonych błędów.
5. Tester wydajnościowy:
   1. Przeprowadza testy wydajnościowe, aby ocenić szybkość i skalowalność strony.
   2. Analizuje wyniki testów i identyfikuje potencjalne źródła problemów z wydajnością.
   3. Współpracuje z zespołem deweloperskim w celu optymalizacji wydajności.
6. Tester bezpieczeństwa:
   1. Przeprowadza testy bezpieczeństwa w celu identyfikacji luk w zabezpieczeniach.
   2. Raportuje o znalezionych podatnościach i proponuje środki zaradcze.
   3. Pomaga w zapewnieniu, że strona jest odporna na ataki zewnętrzne.
7. Administrator testów:
   1. Zarządza narzędziami do testowania i środowiskiem testowym.
   2. Zapewnia konfigurację narzędzi i infrastruktury testowej.
   3. Udziela wsparcia technicznego zespołowi testowemu.
8. Analityk danych:
   1. Analizuje wyniki testów i przygotowuje raporty z testów.
   2. Pomaga w identyfikacji trendów i problemów w wynikach testów.
   3. Współpracuje z zespołem testowym i deweloperskim w celu poprawy jakości produktu.