

# Plan testów dla sklepu internetowego Demoblaze

## 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

## **1. Wstęp**

Niniejszy dokument przedstawia plan testów dla aplikacji webowej **Demoblaze** – sklepu internetowego z elektroniką. Celem testów jest weryfikacja poprawności działania kluczowych funkcjonalności aplikacji, jej stabilności, responsywności oraz zgodności z oczekiwaniami użytkowników. Testy są prowadzone z użyciem frameworka Cypress.

## **2. Zakres testów**

Testy obejmują następujące obszary funkcjonalne aplikacji:

- Działanie strony głównej
- Nawigacja po menu
- Formularze (logowanie, rejestracja, kontakt)
- Koszyk (dodawanie, usuwanie, suma produktów)
- Proces składania zamówienia
- Karuzela na stronie głównej
- Filtrowanie po kategoriach
- Stopka (footer)
- Responsywność na różnych urządzeniach
- Walidacja formularzy
- Obsługa alertów i komunikatów

Typy testów wykluczone z planu testów:

- Testy bezpieczeństwa - brak kompetencji osób w zespole związanych z dziedziną pentestingu.

## **3. Przedmiot testów**

Przedmiotem testów jest aplikacja internetowa dostępna pod adresem:  
<https://www.demoblaze.com/>

## **4. Kryteria zaliczenia**

Test uznaje się za zaliczony, gdy:

- Wszystkie kroki testowe zostały wykonane zgodnie ze scenariuszem
- Rzeczywisty rezultat jest zgodny z oczekiwany

- Nie wystąpiły żadne krytyczne błędy
- Test został wykonany w zaplanowanym środowisku
- Czas odpowiedzi serwera nie może być dłuższy niż 2000 ms

## **5. Kryteria niezaliczenia**

Test uznaje się za niezaliczony, gdy:

- Wystąpił błąd krytyczny uniemożliwiający dalsze testowanie
- Rzeczywisty rezultat różni się od oczekiwanej wartości
- Aplikacja nie odpowiada lub zachowuje się niestabilnie
- Funkcjonalność nie działa zgodnie z dokumentacją

## **6. Kryteria wejścia**

Warunki rozpoczęcia testów:

- Aplikacja jest dostępna pod wskazanym adresem URL
- Środowisko testowe jest skonfigurowane (Node.js, Cypress)
- Scenariusze testowe są zdefiniowane
- Dane testowe są przygotowane
- Zespół testowy posiada wymagane uprawnienia

## **7. Kryteria wyjścia**

Warunki zakończenia testów:

- Wykonano wszystkie zaplanowane przypadki testowe
- Raport z testów został wygenerowany
- Wszystkie znalezione błędy zostały udokumentowane
- Osiągnięto wymagany poziom pokrycia testami

## **8. Lista funkcjonalności do przetestowania**

- Dostępność strony głównej
- Nawigacja po menu
- Formularz kontaktowy

- Rejestracja użytkownika
- Logowanie użytkownika
- Karuzela na stronie głównej
- Kategorie produktów
- Dodawanie do koszyka
- Usuwanie z koszyka
- Suma koszyka
- Modal zamówienia
- Składanie zamówienia
- Walidacja pustego formularza
- Stopka (footer)
- Responsywność na mobile
- Responsywność na tablecie

## **9. Środowisko testowe**

### **Sprzęt:**

- Laptop/PC z systemem Windows 10/11

### **Oprogramowanie:**

- Przeglądarka: Google Chrome (wersja 120+)
- Node.js (wersja 18+)
- Cypress (wersja 13+)
- Visual Studio Code
- Jmeter

### **Konfiguracja:**

- System operacyjny: Windows
- Rozdzielczości testowane: 375x667, 768x1024, 1366x768, 1920x1080

## 10. Kategorie błędów

Kategoria	Opis	Priorytet
Krytyczny	Błąd uniemożliwiający korzystanie z kluczowych funkcji (np. brak możliwości złożenia zamówienia)	P1 - Natychmiastowa naprawa
Wysoki	Błąd wpływający na główne funkcjonalności (np. nie działa dodawanie do koszyka)	P2 - Wysoki priorytet
Średni	Błąd wpływający na funkcje drugorzędne (np. nie działa karuzela)	P3 - Do naprawy w kolejnym wydaniu
Niski	Błąd kosmetyczny, literówki, drobne problemy z UI	P4 - Może być poprawione później

## 11. Miejsce testów

- Testy wykonywane są zdalnie na lokalnym środowisku deweloperskim
- Aplikacja testowana jest poprzez przeglądarkę internetową
- Repozytorium z testami znajduje się na lokalnym dysku:  
C:\Users\aleks\OneDrive\Pulpit\Demoblaze.Cypress\

## 12. Harmonogram testów

Etap	Opis	Data rozpoczęcia	Data zakończenia
1	Smoke testy	10.02.2026	12.02.2026
2	Testy koszyka	13.02.2026	15.02.2026
3	Testy zamówień	15.02.2026	16.02.2026
4	Testy UI i karuzeli	16.02.2026	17.02.2026
5	Testy kategorii	17.02.2026	18.02.2026
6	Testy footer'a	18.02.2026	18.02.2026
7	Testy responsywności	18.02.2026	19.02.2026
8	Raport końcowy	19.02.2026	20.02.2026

### **13. Raporty z testów**

Raporty z testów będą zawierać:

- Liczbę wykonanych testów
- Liczbę testów zakończonych sukcesem
- Liczbę testów zakończonych porażką
- Listę znalezionych błędów z opisem i priorytetem
- Screenshoty z testów
- Czas wykonania testów

Raporty generowane są poprzez:

- Konsolę Cypress podczas uruchamiania testów
- Screenshoty robione automatycznie w testach

### **4. Lista narzędzi**

<b>Narzędzie</b>	<b>Wersja</b>	<b>Zastosowanie</b>
Cypress	13.15.0	Framework do automatyzacji testów
Node.js	20.x	Środowisko uruchomieniowe JavaScript
Visual Studio Code	Latest	Edytor kodu
Git	Latest	Kontrola wersji
GitHub	-	Repozytorium zdalne
Chrome DevTools	-	Debugowanie
Excel/Microsoft 365	-	Dokumentacja testów
Jmeter	-	Testy wydajnościowe
PicPick	-	Narzędzia do tworzenia zrzutów ekranu i filmików

### **15. Zarządzanie incydentami**

**Proces zgłaszania błędu:**

1. Wykonanie testu i zaobserwowanie niezgodności
2. Dokumentacja błędu (opis, kroki reprodukcji, screenshot)
3. Określenie priorytetu (Krytyczny/Wysoki/Średni/Niski)

4. Zapisanie w raporcie końcowym
5. Oznaczenie testu jako "NEGATYWNY" w dokumentacji

#### **Eskalacja:**

- Błędy krytyczne: natychmiastowe zgłoszenie
- Błędy wysokie: do naprawy przed kolejnym wydaniem
- Błędy średnie/niskie: do naprawy w przyszłych wersjach

#### **16. Role i odpowiedzialność**

<b>Rola</b>	<b>Osoba</b>	<b>Odpowiedzialność</b>
<i>Tester Automatyzujący</i>	Aleksandra	Projektowanie i implementacja testów, wykonywanie testów, raportowanie wyników
<i>Analityk Testów</i>	Aleksandra	Tworzenie scenariuszy testowych, analiza wymagań
<i>Lider Projektu</i>	-	Nadzór nad projektem, akceptacja wyników
<i>Developer</i>	-	Naprawa zgłoszonych błędów