Plan testów dla sklepu internetowego “pachnacykociolek”

# Spis treści:

1. Wstęp
2. Przedmiot testów
3. Zakres 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 narzędziami
16. Role i odpowiedzialność

# Wstęp

Celem niniejszego planu testów jest szczegółowe przetestowanie funkcjonalności sklepu w oparciu o nabyte dotychczas doświadczenie. Testowaną stroną jest witryna: pachnacykociolek.pl.

# Przedmiot testów

Przedmiotem testów jest sklep internetowy dostępny pod adresem pachnacykociolek.pl wraz z jego wszystkimi funkcjonalnościami.

# Zakres testów

Poziomy testów objęte w planie to:

* Testowanie modułowe
* Testowanie integracyjne
* Testowanie akceptacyjne

## Typy testów ujęte w planie testów:

* Testy funkcjonalne - oprogramowanie zostanie zweryfikowane pod kątem typowych funkcjonalności dla tego typu stron
* Testy integracyjne - testowanie komunikacji pomiędzy API
* Testy automatyczne - implementacja testów automatycznych do kluczowych funkcjonalności systemu na podstawie dostarczonych przypadków testowych

## Typy testów wykluczone z planu testów:

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

# Kryteria zaliczenia

## Testy wydajnościowe

* Czas odpowiedzi serwera nie może być dłuższy niż 1000ms

## Testy funkcjonalne

* Kryteria zaliczenia są zgodne z obecną implementacją sklepu internetowego

## Testy automatyczne

* Zostały zautomatyzowane wszystkie przypadki testowe
* Testy zostały podłączone do narzędzia CI/CD - Jenkins

# Kryteria niezaliczenia

## Testy wydajnościowe

* Czas odpowiedzi serwera jest dłuższy niż 1000 ms

## Testy funkcjonalne

* Zaimplementowane funkcjonalności nie są zgodne z typowymi funkcjonalnościami dla tego typu stron

# Kryteria wejścia

* Uruchomione środowisko testowe
* Dostęp do urządzenia z systemem IOS

# Kryteria wyjścia

* Wykonane wszystkie przypadki testowe
* Wszystkie typy i poziomy testów uwzględnione w planie testów zostały zrealizowane
* poprawione zostały wszystkie defekty

# Lista funkcjonalności do przetestowania

* Wyszukiwarka produktów - przypadek testowy 001
* Proces składania zamówienia - przypadek testowy 002
* Tworzenie produktu - przypadek testowy 003
* Panel administracyjny - przypadek testowy 004

# Środowisko testowe

Testy sklepu internetowego będą przeprowadzane na dedykowanym serwerze pachnacykociolek.pl. Środowisko testowe jest zbliżone parametrami do środowiska produkcyjnego.

## Specyfikacja:

Procesor: 2GHz

RAM: 4GB

DYSK: 240 GB

System: Linux Debian 17

## Testy będą przeprowadzane przy użyciu poniższych urządzeń:

1. Komputer PC

* Procesor: i7-6700HQ CPU 2,6 Ghz
* RAM: 8GB

1. Urządzenie mobilne z systemem IOS Iphone 14 Pro

* Procesor Apple A16 Bionic
* Pamięć Ram: 6GB
* IOS wersja 16

# Kategorie błędów

Wyróżniamy różne priorytety w naszym planie testów:

* Trywialny
* Średni
* Poważny
* Krytyczny

# Miejsce testów

Testy będą przeprowadzane w siedzibie firmy wytwarzającej oprogramowanie przez testera.

# Harmonogram testów

1. Testowanie statyczne

* Weryfikacja przygotowanych dokumentacji
* Weryfikacja historyjek użytkownika

czas 8h

## Testy funkcjonalne

* Weryfikacja zaimplementowanych funkcjonalności zgodnie z funkcjonalnościami dla tego typu stron
* Raportowanie defektów
* Wykonanie przypadków testowych

czas 30h

## Testy wydajnościowe

* Weryfikacja czasu odpowiedzi serwera

czas 10h

## Testy automatyczne

* Przygotowanie struktury plików
* Implementacja testów automatycznych w oparciu o utworzone przypadki testowe
* Podłączenie testów pod narzędzie Jenkins
* Podłączenie testów pod narzędzie Grafana

czas 50h

# Raporty z testów

* Raporty o wykrytych defektach
* Zaprojektowane przypadki testowe
* skrypty testów automatycznych
* Metryki testów wydajnościowych

# Lista narzędzi

1. Narzędzia do raportowania defekatów

* Jira

## Narzędzia do tworzenia przypadków testowych

* TestLink
* XRay

## Narzędzia do automatyzacji

* WebdriverIO
* SuperTest
* Allure
* Jenkins
* Grafana

## Narzędzia do testów wydajnościowych

* Jmeter

## Narzędzia do tworzenia zrzutów ekranu i filmików

* PickPick
* ShareX

# Zarządzanie incydentami

Po wykryciu każdy defekt zostanie zgłoszony do systemu zarządzania defektami (Jira). Taki defekt będzie posiadał priorytet wskazany w planie testów oraz osobę, która będzie zajmowała się poprawą błędu.

Gdy błąd zostanie poprawiony będzie przekazany testerowi do przeprowadzenia testów potwierdzających.

Po poprawnie naprawionym defekcie, zgłoszenie zostanie zamknięte i przesunięte do zakładki gotowe.

W przypadku nieprawidłowej naprawy defektu, zgłoszenie na nowo zostanie przekazane do programisty i tak aż do poprawnego wyeliminowania defektu.

# Role i odpowiedzialność

Kierownik testów - nadzorowanie i organizowanie pracy całego zespołu testowego

Analityk testów - analizowanie podstawy testów. Przygotowanie testaliów potrzebnych do testów

Tester - Realizuje zadania wyznaczone przez kierownika testów, wykonuje testy i raportuje defekty

Administrator techniczny systemu - przygotowuje środowisko testowe i je nadzoruje podczas procesu testowego.