QA Manager – techniczne zadania

Zadanie 1 – Szybki test do monitoringu
Celem zadania jest opracowanie i napisane testu automatycznego opartego o API (graphql + REST), który będzie podstawą monitoringu ścieżki krytycznej sklepu https://www.castorama.pl/

Należy pamiętać że jest to test który chcemy wpiąć do monitoringu strony, a więc duży nacisk stawiany jest na: stabilność, szybkość wykonania i elementy które są sprawdzane (asercje).

Techniczne pomoce:
* definicje graphql powinny zawierać tylko niezbędne dane
* test powinien się skończyć przed złożeniem fizycznie zamówienia (z uwzględnieniem akcji na checkout / cart)

Zadanie 2 – Test funkcjonalny sklepu
Celem zadania jest opracowanie koncepcji i napisanie działającego szkieletu testów automatycznych ścieżki krytycznej sklepu https://semilac.pl/

W tym zadaniu najważniejsze jest zastosowane podejście i umiejętność rozplanowania repozytorium które będzie utrzymywane i łatwo rozwijalne.
Testujemy funkcjonalności sklepu, a więc warto aby scenariusz / przypadki testowe w ramach samych testów były zapisane w sposób przejrzysty i zrozumiały np. dla osób z biznesu, a same testy spełniały wymagania klienta co do dostępności strony na wspieranych przeglądarkach.

Techniczne pomoce:
* sklep jest wielojęzykowy, warto już na poziomie tworzenia szkieletu pamiętać o tym i rozplanować możliwości testowania innych witryn niż PL
* test powinien się skończyć przed złożeniem fizycznie zamówienia (z uwzględnieniem akcji na checkout / cart)

Zadanie 3 – Konteneryzacja i raport
Do napisanego testu w zadaniu 2 należy dodać definicję konteneryzacji rozwiązania, np. Docker. Ważne aby dzięki temu zadaniu można było odpalić testy na dowolnym urządzeniu bez potrzeby instalowania czegoś więcej niż usługa potrzebna do zbudowania / odpalenia kontenera.

Wynikiem przebiegu odpalenia kontenera powinien być raport z przebiegu testu.

Zadanie 4 – Integracja z narzędziem Cloudowym
Do testu z zadania 2 należy dodać opcję odpalenia go w dowolnej chmurze dostarczającej farmę urządzeń, np. BrowserStack (ma bezpłatną opcję trial)

Oddanie zadań:

Same zadana powinny być oddane w wersji pozwalającej umieścić je w repozytorium (a więc bez zbędnych elementów takich jak zaciągnięcie zależności, cache IDE itp.).

a) zadanie 1 powinno zawierać repozytorium / archiwum skryptu(ów) wraz z dokumentem (np. dla opsów, działu monitoringu) sugerującym jakie elementy strony są testowane oraz instrukcję jak odpalić dany skrypt wraz z określeniem częstotliwości odpalania skryptu,
b) zadanie 2 i 3 repozytorium / archiwum – wraz ze wszystkimi plikami potrzebnymi do odpalenia oraz instrukcją, zasadami dobrych praktyk użytych w kodzie,
c) w ramach zadania nr 4 należy przesłać dane dostępowe w konfiguracji testów do danej chmury, a także dane autoryzacyjne które pozwolą przejrzeć przebieg testu.

Wszystkie zadania mogą być oddane w ramach jednego dużego repozytorium z podziałem na poszczególne dwa rodzaje testów.