

**Szczegółowy opis przedmiotu zamówienia**

**dot. zmian dla strony www Culture.pl wraz z submarkami (subdomenami) oraz stroną IAM.PL**

Specyfikacja techniczna wraz z dokumentacją funkcjonalną Culture.pl (Załącznik nr 10) definiuje szereg wymagań stawianych wobec wykonawcy odnośnie wdrażania dowolnej zmiany w zakresie serwisu głównego, bądź serwisów powiązanych. Poniżej znajduje się opis tych wymagań, z których kluczowe to:

- responsywność
- zgodność z WCAG

Szczegóły związane z wymaganiami odnośnie SLA znajdują się w dokumentacji funkcjonalnej znajdującej się w załączniku nr 10 do SIWZ oraz w załączniku nr 9 do SIWZ – SLA, kategoryzacja błędów, procedury obsługi awarii.

**Responsywność**

Wszelkie zmiany powinny być przygotowane tak, aby strona była responsywna, to znaczy, że wszystkie strony i elementy muszą być pocięte i wdrożone pod następujące szerokości ekranu:

1. 1280px
2. 1024px
3. 960px
4. 720px
5. 640px
6. 480px

Co oznacza, że elementy interfejsu powinny być oparte na siatce, pozwalającej dostosować je do każdej z powyższych progowych szerokości ekranu. W chwili obecnej korzystamy z siatki 20 kolumn oraz frameworka Bootstrap i to nie może ulec zmianie oraz nie powinno być ignorowane - wszelkie cięcie musi się odbywać w ramach tego frameworka CSS.

**Brak nadmiarowych elementów HTML**

Ze względu na wybraną technologię, występuje bardzo duże ryzyko powstawania bardzo zagnieżdżonej struktury DOM, ze względu na to, że standardowe funkcjonalności Drupala generują tego typu strukturę automatycznie. Wymagamy, aby projekt był wykonywany tak, aby ograniczyć liczbę elementów HTML do absolutnego minimum pozwalającego oscylować dany element. Dopuszcza się nadmiarowe elementy jedynie w przypadku, gdy inne moduły używane w systemie wymagają ich do pracy, czyli pozbycie się ich wymagałoby przepisania także tych modułów.

**Odbiór elementów frontendu**

Odbiór graficznych elementów serwisu będzie polegał na przetestowaniu ich przez Zleceniodawcę w związku, z czym Wykonawca jest zobowiązany do przygotowania testów dla Zamawiającego, pozwalających na:

- przetestowanie rozwiązania w każdej możliwej rozdzielczości obsługiwanej przez serwis
- przetestowanie każdego z możliwych stanów danego elementu (aktywny, nieaktywny, kliknięty, odwiedzony)
- przetestowanie każdej z możliwych sytuacji w zakresie uzupełnienia funkcjonalności treścią:
  - dla krótkich i długich tytułów
  - w sytuacji gdy brakuje zdjęcia
  - w sytuacji gdy lead jest krótki/długi
  - etc.
- przetestowanie rozwiązania zarówno na specjalnie w tym celu przygotowanej stronie, oraz na istniejących

podstronach serwisu

- specjalna strona służąca do testowania powinna osadzać daną funkcjonalność obok wszelkich innych funkcjonalności dostępnych w serwisie, aby móc w prosty sposób wykryć ewentualne konflikty
- przykładem może być podpis pod zdjęciem gdy pod nim znajduje się cytat, galeria etc.

## **Dokumentacja**

Wykonawca powinien przedstawić szczegółową dokumentację wykonanych zmian.

Dokumentacja powinna się składać z:

1. Specyfikacji Funkcjonalnej szczegółowo opisującej sposób wdrożenia danych rozwiązań określonych przez Zamawiającego, z rozbiciem na:
  - a. sekcję frontendową, zawierającą opis:
    - i. każdego z ekranów zaprojektowanych na poziomie UX i narysowanych przez grafika
    - ii. każdej z funkcjonalności wraz z opisem jej zasady działania, w tym m.in:
      1. skąd pobierana jest zawartość danej sekcji/widgetu
      2. w jaki sposób jest ona sortowana
      3. czy ma ona ograniczenia/filtrowanie
  - b. sekcję backendową, zawierającą opis:
    - i. dodawania każdego typu treści, wraz ze szczegółowym opisem pokazującym, w jaki sposób można osiągnąć wszelkie, zaprojektowane i narysowane wcześniej układy treści i multimediów - w tym także mechanizmami promocji treści, takimi jak karuzele, popupy, widgety etc.
    - ii. zarządzanie innymi elementami systemu, wymaganymi przez Zleceniodawcę, takimi jak zarządzanie użytkownikami, zarządzanie dostępem do stron, historią zmian, dziennikiem systemowym, włączania/wyłączania danych funkcjonalności
2. Specyfikacji Technicznej składającej się z:
  - a. szczegółowego opisu wymagań technicznych aplikacji zawierającej informacje o wszelkich, wymaganych przez aplikację usługach, programach i bibliotekach, wraz z ich wersjami (PHP 5.3, java, biblioteki do konwersji grafiki, biblioteki kryptograficzne, rozszerzenia PHP etc.)
  - b. szczegółowej, wygenerowanej automatycznie z kodu źródłowego aplikacji, dokumentacji w postaci dokumentu elektronicznego

Wykonawca ma obowiązek aktualizować dokumentację podczas wprowadzania zmian, bądź poprawek w systemie, także po wykonaniu zlecenia, w okresie trwania gwarancji.

Wykonawca powinien prowadzić dokumentację zmian w repozytorium wykorzystanym podczas tworzenia aplikacji. Zamawiający powinien mieć pełen dostęp, co najmniej do gałęzi produkcyjnej aplikacji w repozytorium, z możliwością podejrzenia zmian oraz komentarzy dotyczących tych zmian.

## **Wydajność Aplikacji**

Serwis powinien używać cache dla wygenerowanych stron, oraz zapytań do bazy danych.

CMS powinien umożliwiać z poziomu panelu administracyjnego, wyczyszczenie każdego poziomu cache, na wypadek zaistnienia takiej potrzeby.

W momencie aktualizacji danego artykułu, cache widoków, dla których wyświetla się artykuł powinien być czyszczony - aplikacja powinna czyścić także widok strony głównej lub kategorii, jeśli na tych stronach pokazuje się zająwka artykułu.

## Frontend

Skórka serwisu powinna zostać wykonana zgodnie z aktualnymi najlepszymi praktykami dotyczącymi wydajności aplikacji pod kątem czasu ładowania się strony.

W szczególności:

- wszelkie grafiki wykorzystywane jako elementy interfejsu, powinny być połączone w jeden "sprite"
- ikony lub inne znaki graficzne powinny być połączone w plik czcionki (woff, ttf) i serwowane, jako jeden zasób
- małe obrazki, będące elementami interfejsu, powinny być zaszyte w plikach css w postaci ciągu data: image base64
- wszelkie zasoby statyczne, będące częścią interfejsu graficznego (css, js, ikony, czcionki, obrazki interfejsu) powinny być serwowane z osobnej subdomeny
- obrazki niebędące elementami interfejsu powinny być ładowane z wykorzystaniem mechanizmu "lazy load" - co oznacza ich załadowanie dopiero w momencie, gdy użytkownik przewinie stronę w przeglądarce do miejsca, w którym powinna się wyświetlić dana grafika.
- pliki javascript i css powinny być łączone w jeden plik i kompresowane
- biblioteki javascript, takie jak jquery, powinny być pobierane z adresów CDN Google/Microsoft - jako jedyne są wyłączone z obowiązku łączenia i kompresji
- Ze względu na RFC2616, to aplikacja steruje nagłówkami expires, w związku, z czym, obrazki oraz inne elementy statyczne, powinny mieć ustawiony maksymalny czas przedawniania
- strony powinny wysyłać nagłówki ETag aby umożliwić weryfikację, czy wersja znajdująca się w cache przeglądarki jest aktualną wersją strony
- wszelkie skrypty powinny być ładowane asynchronicznie, a układ strony i podstawowe definicje elementów powinny zapobiegać "skakaniu" layoutu, jeśli któryś ze skryptów zmienia jego wygląd/położenie elementów
- reguły CSS oraz modyfikacje w drzewie DOM wykonywane poprzez skrypty javascript, powinny powodować minimalną liczbę zdarzeń typu reflow w przeglądarce
- strona może wykorzystywać framework css
- strona powinna zawierać minimalną ilość kodu html, tak, aby proporcja treści strony do kodu wyświetlającego tą treść była jak największa
- szczególną uwagę należy tutaj zwrócić na generowanie nadmiarowych klas i identyfikatorów, oraz wielokrotnie zagnieżdżonych znaczników, przez system CMS
- obrazki wysyłane przez serwer nie powinny być kompresowane przez serwer (gzip) a jedynie optymalizowane przez aplikację w momencie generowania miniatur.

## Backend

Czas odpowiedzi strony html (mediana) nie powinien przekraczać

- 2000ms przy "zimnym" cache,
- 600ms przy wykorzystaniu cache aplikacji i
- 250ms przy wykorzystaniu Varnish cache.

Powyższe czasy zakładają, że na stronie znajduje się jednocześnie 500 użytkowników oraz wykonywane jest 50 odsłon stron na sekundę. Odsłona oznacza tutaj pobranie strony wraz z wszystkimi zasobami wymaganymi do jej wyświetlenia.

Dla wersji mobilne strona powinna dynamicznie generować mniejsze obrazki, dostosowane do wymiarów strony, aby nie obciążać użytkowników mobilnych ani na poziomie przesyłu danych, ani poprzez zbędne zajmowanie pamięci ich urządzenia.

Funkcje takie jak ocenianie artykułów, powinny być zaimplementowane za pomocą mechanizmów Ajax. Samo

wyświetlanie powinno być wbudowane w stronę, ale aktualizacja oceny po zgłoszeniu powinna być wysyłana do serwera za pomocą żądania javascript.

### Testy wydajności

Wykonawca powinien dostarczyć wraz z projektem, wyniki badania wydajności aplikacji za pomocą narzędzia klasy ab/siege/jmeter.

Raport powinien zawierać:

- dokładne informacje o konfiguracji sprzętowej serwera, na którym była testowana aplikacja, konfiguracji poszczególnych usług (serwer www, baza danych, wszelkie mechanizmy wykorzystane do developowania strony)
- dokładnej konfiguracji narzędzia, którym wykonywano pomiar
  - ile wątków wykorzystywało narzędzie
  - jaka była przepustowość łącza między serwerem testowym a serwerem z aplikacją testującą
- informacje o wynikach testu dla aplikacji w stanach
  - wyłączony cache
  - włączony cache
  - rozgrzany cache
  - zimny cache
- szacunki dotyczące ograniczeń aplikacji i poziomu ruchu jaki jest w stanie obsłużyć przy wykorzystaniu proponowanej konfiguracji sprzętowej
- zmiany, które zostały wprowadzone w konfiguracji usług po testach wydajnościowych

Testy wydajności powinny zawierać informacje zarówno o średnim czasie odpowiedzi i pobierania strony, jak i inne statystycznie istotne miary, takie jak mediana czy odchylenie standardowe dla żądań.

Testy dotyczące tego punktu mogą być wykonywane z założeniem, że proporcja odczytów do zapisów cache jest na poziomie 9/1.

Czas reakcji nie uwzględniają opóźnień wynikających z przesyłu danych przez sieć internet - chodzi wyłącznie o czas generowania kodu - nie jego przesyłania, w związku, z czym testy mogą być wykonywane w obrębie sieci lokalnej.

### SEO

Wszelkie zmiany powinny być wprowadzane zgodnie z aktualnymi najlepszymi praktykami SEO. Jakiegokolwiek zmiany w strukturze html powinny być konsultowane z Zamawiającym.

### Analitka internetowa

Culture.pl oraz inne serwisy, których dotyczy Umowa posiadają wdrożone zaawansowane śledzenie zachowań użytkowników za pomocą Google Tag Manager oraz Google Analytics. Jest bardzo prawdopodobne, że wprowadzane zmiany i wykonywane funkcjonalności będą wymagały zmian także na poziomie implementacji analityki.

### Bezpieczeństwo

Wykonawca ponosi odpowiedzialność za utrzymywanie systemu w aktualnej wersji. Aktualizacja systemu w zakresie poprawek bezpieczeństwa powinna się odbywać najpóźniej na 4 godziny od momentu publikacji informacji o luce bezpieczeństwa.

Jednocześnie Wykonawca jest w pełni odpowiedzialny za bezpieczeństwo modułów wykonywanych w całości przez siebie, bądź modyfikacji wprowadzanych do oficjalnych modułów. Wyjątkiem od tej sytuacji są luki bezpieczeństwa występujące w wyniku luk w mechanizmach core drupala, w przypadku, gdy Wykonawca stosował wszelkie zalecenia dotyczące bezpieczeństwa w swoim module.

### **Gwarancja**

Wykonawca będzie odpowiedzialny za bezpłatne aktualizacje oraz rozwiązywanie problemów związanych z działaniem aplikacji, których przyczyna leży po stronie wykonawcy, w okresie 12 miesięcy od momentu wykonania Dzieła. Szczegółowe warunki SLA są zawarte w załączniku nr 9 do SIWZ.

Zmiany wynikające z woli Zleceniodawcy, będą każdorazowo wyceniane przez Wykonawcę i wykonywane w drodze osobnych ustaleń.

Oferta powinna zawierać cennik godzinowy Wykonawcy.

### **Wersja mobilna**

Serwisy Zamawiającego nie posiadają wersji mobilne i nie jest dopuszczalne stosowanie rozwiązań, które omijają temat responsywności poprzez inną prezentację treści wyłącznie dla urządzeń mobilnych. Wyjątkiem od tej reguły jest wyraźne polecenie ze strony Zleceniodawcy stwierdzające, że dany serwis lub podstrona ma być dostępna w takiej wersji.

### **Wersja dla niepełnosprawnych**

Wszelkie modyfikacje powinny być wykonane w taki sposób, aby nie była wymagana osobna wersja dla osób niepełnosprawnych.

Serwis powinien w pełni implementować standard WCAG 2.0 co najmniej na poziomie AA - dotyczy to wszelkich zmian, poprawek oraz nowych funkcjonalności.

### **Wersja żałobna serwisu**

Serwis zawiera specjalny zestaw styli dla wersji żałobnej serwisu, którą można będzie włączyć z poziomu panelu administracyjnego systemu. Wszelkie modyfikacje powinny uwzględniać tę funkcję i powinna być dostępna wersja żałobna umożliwiająca testowanie zmian z włączoną tą opcją.

Wersja żałobna zmienia także paletę kolorów dla obrazków wyświetlanych na stronie.

### **Wersja językowa serwisu**

System jest w całości zaimplementowany w języku polskim, co oznacza, że zarówno elementy frontendu, jak i backendu, powinny być po polsku. Wykonawca jest zobowiązany przygotować każdą funkcjonalność w języku polskim oraz angielskim - włącznie z wprowadzeniem tłumaczeń dla elementów maski serwisu.

Jednocześnie, ze względu na problemy z tłumaczeniami w przypadku gdy standardowy język CMS Drupal7 jest ustawiony na język inny, niż angielski, system powinien mieć ustawiony jako podstawowy język angielski, co oznacza, że wszelkie tokeny i teksty zaszyte w modułach oraz szablonach, powinny być napisane po angielsku.

### **Panel administracyjny**

Wszelkie zmiany w polach w formularzy służących do tworzenia treści bądź edycji struktury serwisu, powinny

mieć wbudowaną pomoc w trybie "inline" - czyli pojawiające się w dymkach podpowiedzi dokładnie opisujące, co należy wprowadzić w danym polu.

Dokumentacja pól nie może być tylko i wyłącznie zawarta w osobnej sekcji systemu.

Wszelkie pola formularza powinny mieć ustawione konkretne typy i obsługiwać walidację zarówno na poziomie javascript, jak i po stronie systemu.

Dokumentacja powinna być wprowadzona zarówno po polsku jak i po angielsku.

### **Właściwości systemu CMS**

Serwisy IAM są zbudowane w oparciu o system Drupal w wersji 7.

- Kod wszelkich wykorzystanych wtyczek powinien być dostępny na zasadach Open Source (licencja typu GPL).
- Nie jest dopuszczalne wykorzystanie komercyjnych wtyczek, lub wtyczek, które w wersji otwartej/darmowej realizują jedynie część funkcjonalności dostępnych w płatnej wersji wtyczki
- Wszelkie moduły/pluginy wykorzystane przy wytwarzaniu serwisu, powinny wykorzystywać warstwę abstrakcji dostarczaną przez CMS
- Moduły powinien korzystać z najbardziej popularnej na rynku biblioteki javascript - jQuery
- Moduły powinny wykorzystywać wewnętrznych systemów ograniczających dostęp dla uprawnionych użytkowników.
- Wszelkie zmiany wprowadzane w systemie powinny być przechowywane w postaci „Features” aby umożliwić kontrolę zmian konfiguracji znajdującej się w bazie danych oraz wymusić zależności konkretnych funkcjonalności od modułów zewnętrznych.

### **Zakres obsługi SLA**

Zamawiający przewiduje, że w ramach SLA będzie wymagał ok 30 godzin pracy Wykonawcy miesięcznie w okresie trwania Umowy (12 miesięcy od podpisania umowy).

### **Zakres obsługi CR**

Zamawiający przewiduje, że w ramach zmian w serwisie będzie wymagał ok **1700** (tysiąc siedemset) godzin pracy Wykonawcy w okresie trwania umowy.

....., dnia .....

.....  
uprawniony przedstawiciel wykonawcy