Wymagania projektowe

1. i 2. etap projektu - Biznes Elektroniczny

Spis treści

Wprowadzenie

Celem projektu jest sprawdzenie Państwa praktycznej wiedzy z zakresu stosowania, konfiguracji i wdrożenia oprogramowania Open Source dedykowanego do tworzenia sklepów internetowych. W ramach zajęć będą Państwo korzystać z oprogramowania PrestaShop i dostosowywać go do określonych wymagań. W drugim etapie projektu będą Państwo musieli przygotować rozwiązanie do wdrożenia i wdrożyć je w dedykowanym środowisku docelowym.

Projekt ma za zadanie pokazać Państwu warunki pracy programisty zbliżone do rzeczywistych od projektu rozwiązania, poprzez dostosowanie produktu dla klienta, współpracę w zespole programistycznym oraz zadania związane z końcowym przygotowaniem produktu do uruchomienia produkcyjnego. Projekt celowo realizowany jest z użyciem istniejących na rynku bibliotek i programów, ponieważ celem na rynku jest jak najszybsze dostarczenie produktu klientowi.

W ramach zadań zaplanowanych do realizacji nauczą się Państwo m.in.:

1. Wykorzystania repozytorium kodu źródłowego do zarządzania i organizacji pracy w zespole
2. Pracy z rzeczywistym rozwiązaniem i cudzym kodem źródłowym oraz prac z dokumentacją techniczną
3. Umiejętności stosowania wirtualizacji oraz konteneryzacji podczas wykonywania prac developerskich oraz wdrożeniowych
4. Pracy z serwerami i wykorzystania orkiestratora kontenerów
5. Wykorzystania połączeń VPN oraz tuneli SSH
6. Stosowania bibliotek automatyzujących testy integracyjne oraz testy UI
7. Dostosowywania wyglądu aplikacji internetowych do wymagań klienta
8. Bibliotek i narzędzi do automatycznego pobierania dużych zbiorów informacji ze stron internetowych
9. Platformy do analizy ruchu internetowego Google Analytics, podstaw jej konfiguracji oraz metod integracji z docelową usługą.

Do realizacji zadań projektowych będą Państwo potrzebować następującego oprogramowania:

- Repozytorium kodu źródłowego - https://github.com lub https://gitlab.com
- Narzędzie do zarządzania zmianami w kodzie - https://git-scm.com
- Platforma sklepu internetowego - **Prestashop 1.7.8** - https://github.com/PrestaShop/PrestaShop/tree/1.7.8.x
- Dowolny edytor kodu źródłowego np. NeoVim/SublimeText/Visual Code - https://code.visualstudio.com
- Serwer maszyn wirtualnych np. VirtualBox https://www.virtualbox.org lub inny
- System konteneryzacji aplikacji - Docker https://docker.com
- System do zarządzania kompozycją kontenerów - https://docs.docker.com/compose/
- Biblioteka do budowy testów UI - https://www.selenium.dev/documentation/en/
- Obraz systemu operacyjnego Ubuntu LTS - https://ubuntu.com/download/desktop
- Podczas konfiguracji środowiska wirtualnego będą przydatne dodatkowe pakiety np.: apache (httpd2), mysql (mariadb), git, php, php-mysql, php-pdo, i inne wskazane

przez platformę do tworzenia sklepów internetowych (skrypt instalacyjny wskaże brakujące biblioteki).

Bardzo ważne jest, aby zaraz po zapoznaniu się z niniejszym dokumentem zaplanowali Państwo harmonogram i podział prac w zespole. W poprzednich latach głównym powodem braku zaliczenia zajęć projektowych były opóźnienia w rozpoczęciu prac.

Proces zgłaszania zespołów

Zgłoszenie zespołu należy wykonać w arkuszu kalkulacyjnym udostępniony w chmurze:
Lista zespołów.xlsx

1. Każdy zespół wpisuje się na listę uczestników zajęć projektowych (link w systemie e-nauczanie) do dnia 14.10.2024 r.
2. Każdy zespół musi wypełnić kolumny: nr indeksu, nazwa zespołu, adres repozytorium kodu, adres sklepu źródłowego.
3. Nazwa zespołu musi być unikalna w obszarze arkusza (kto pierwszy ten lepszy). Nazwy niecenzuralne, obraźliwe nie będą akceptowane.
4. Adres sklepu źródłowego również musi być unikalny (kto pierwszy ten lepszy).
5. Zespoły mogą liczyć do 5 osób (najmniejsza zalecana liczba studentów w zespole to 3 osoby). Osoby z tego samego zespołu proszę, aby były wpisane w wierszach sąsiadujących.
6. Po 14.10.2024 r. arkusz zostanie udostępniony wyłącznie do odczytu i nie będzie już możliwości wprowadzenia zmian.
7. Osoby, które nie wpiszą się do arkusza lub zespoły, które nie uzupełnią kompletu danych (patrz pkt 2.) nie otrzymają zaliczenia z zajęć projektowych

Wymagania na 1. etap (9-10. godzina zajęć projektowych)

1. **(5 pkt)** *Zespół zakłada repozytorium kodu (projekt) na GitHub/GitLab
    a. Wersjonowany jest tylko kod źródłowy, bez cache, uploadów, itd. – dopuszczalne jest umieszczenie zdjęć zarejestrowanych produktów
    b. W ustawieniach projektu zarejestrowano wszystkich członków zespołu;
    c. Repozytorium zawiera plik .gitignore określający zasoby niepodlegające wersjonowaniu;
    d. Repozytorium zawiera plik README.md z opisem projektu, wykorzystanej wersji oprogramowania, sposobu uruchomienia i składu zespołu.
    e. W gałęzi głównej (master/main) przechowywana jest tylko wersja produkcyjna tzn. prezentowana podczas zaliczenia; **Prowadzący nie ma obowiązku oglądania oprogramowania z innych gałęzi niż master/main.**
    f. W gałęziach pobocznych przechowywane są wersje robocze projektu;
    g. Wprowadzanie zmian do gałęzi głównej master/main (poza inicjalnym commitem) jest możliwe jedynie poprzez wystawienie Pull Request (PR) lub Merge Request (MR) i akceptację tej zmiany przez innego członka zespołu.
    h. Zawartość repozytorium jest podzielona na następujące foldery:
        i. kody źródłowe sklepu;
        ii. kody źródłowe testów automatycznych;
        iii. kody źródłowe narzędzia do scrapowania;
        iv. rezultat scrappowania;
        v. pliki konfiguracyjne i skrypty wykorzystywane w procesie instalacji/ wdrożenia.
    i. Historia pracy zespołu jest prawidłowo rejestrowana w projekcie. Oznacza to m.in., że zadania realizowane przez zespół są opisane jako issues i przypisane do osób odpowiedzialnych za realizację. Realizacja issue powinna być zakończona wystawieniem PR/MR i akceptacją propozycji rozwiązania przez innego członka zespołu.
2. **(5 pkt)** *Zespół utworzył skrypt/program to scrapowania (automatycznego odczytania i zapisania lokalnie informacji/danych z wybranego sklepu internetowego).
    a. Kod źródłowy rozwiązania znajduje się w repozytorium;
    b. Rezultaty scrappowania (kodowanie UTF-8) znajdują się w folderze w repozytorium,
    c. Rezultatami scrappowania są m.in. listy kategorii, podkategorii, produktów (nazwa, opis, cena, atrybuty (np. kolor, materiał), dwa zdjęcia w rozdzielczości która umożliwia wyświetlenia ostrego zdjęcia po powiększeniu w sklepie - miniatury nie są akceptowane)
3. **(5 pkt)** *Zespół utworzył sklep wykorzystując rozwiązanie bazowe **Prestashop w wersji 1.7.8**
    a. Środowisko do uruchomienia sklepu zostało opracowane z wykorzystaniem dedykowanej maszyny wirtualnej z systemem Ubuntu lub zestawu skonteneryzowanych usług zarządzanych za pomocą rozwiązania Docker/ docker-compose.
    b. Sklep pozwala na obsługę żądań i zwraca poprawne odpowiedzi;
    c. Sklep posiada menu górne, stopkę, wyszukiwarkę, umożliwia logowanie i rejestrację, prezentuje listę produktów i umożliwia ich wybór oraz zakup,

dostępne są strony O sklepie, Formy płatności i inne standardowo oferowane strony, dostępne zaraz po instalacji. Możliwe jest przeprowadzenie procesu rejestracji, logowania oraz dostęp do historii zamówień.

d. Prezentowany sklep nie posiada domyślnych wartości zdefiniowanych przez kreator w miejscach prezentowanych klientowi.
e. Interfejs użytkownika jest w języku polskim.
f. W sklepie zarejestrowano conajmniej 1000 produktów (chyba że wybrany sklep źródłowy nie oferuje takiej liczby produktów). Część produktów ma być niedostępna. Stany magazynowe proszę ustawić tak, aby liczba dostępnych produktów każdego rodzaju nie przekraczała 10 szt.
g. Nazwy, opisy, zdjęcia, ceny i podatki określone dla każdego produktu muszą odpowiadać rzeczywistym danym pobranym ze sklepu źródłowego. Polskie znaki muszą się prawidłowo wyświetlać.
h. W sklepie zarejestrowano conajmniej 4 kategorie i w każdej z nich conajmniej 2 podkategorie. Żadna z podkategorii nie może być pusta.
i. W sklepie zdefiniowano metody płatności stosowane w Polsce (nie należy przeprowadzać żadnych integracji do zewnętrznych operatorów płatności). Nie mogą występować metody płatności, których nie stosuje się w Polsce.
j. Poprawnie skonfigurowano mechanizm wysyłania powiadomień e-mail ze sklepu. Przykładowo klient powinien otrzymać wiadomość po rejestracji lub po dokonaniu zakupów.
k. Oferowana w sklepie funkcjonalność działa prawidłowo - tzn. nie generuje błędów (białe strony, 500, 404, 403). Dotyczy to zarówno kodu w plikach PHP, JS jak i referencji do obrazków.
l. Sklep obsługuje żądania jedynie poprzez zastosowanie szyfrowania (HTTPS), za pomocą samodzielnie wygenerowanego i samodzielnie podpisanego certyfikatu SSL.
m. Należy zdefiniować dwóch nowych przewoźników. Należy dostosować ustawienia przewoźników, tak aby dla zamówień powyżej 2000 zł dostawa powinna być darmowa. Dla produktów cięższych niż 50 kg dostawa powinna być niemożliwa. Opłaty dla przewoźników powinny być różne.

4. **(5 pkt)** *Zespół utworzył skrypt do automatycznego testowania sklepu (w technologii Sellenium https://www.selenium.dev/documentation/en/). Skrypt powinien wykonać wszystkie opisane poniżej czynności w czasie do 5 minut. Skrypt pozwala automatycznie wykonać następujące czynności:
    a. Dodanie do koszyka 10 produktów (w różnych ilościach) z dwóch różnych kategorii,
    b. Wyszukanie produktu po nazwie i dodanie do koszyka losowego produktu spośród znalezionych
    c. Usunięcie z koszyka 3 produktów,
    d. Rejestrację nowego konta,
    e. Wykonanie zamówienia zawartości koszyka,
    f. Wybór metody płatności: przy odbiorze,
    g. Wybór jednego z dwóch przewoźników,
    h. Zatwierdzenie zamówienia,
    i. Sprawdzenie statusu zamówienia.

   j. Pobranie faktury VAT.
5. **(7 pkt)**** Interfejs użytkownika w utworzonym sklepie internetowym jest identyczny lub zbliżony (nie obejmuje dostosowania kolorystyki) jak w sklepie źródłowym.
6. **(2 pkt)**** Modyfikacja standardowego wyglądu sklepu:
   a. Wygląd sklepu i sposób prezentacji informacji jest zgodny z oferowanym asortymentem lub jest zgodny ze sklepem bazowym (wskazanym jako źródło danych). Layout sklepu i elementy interfejsu użytkownika są wykonane estetycznie, zgodnie z obowiązującymi standardami.
   b. Na stronie głównej należy dodać co najmniej jeden działający baner w tematyce sklepu,
7. **(5 pkt)** Zespół utworzył skrypt, który poprzez API REST dostępne w oprogramowaniu, zainicjuje automatycznie kategorie, podkategorie, produkty w sklepie na podstawie rezultatów scrapowania,
8. **(3 pkt)** Modyfikacja katalogu produktów:
   a. W katalogu produktów zdefiniowano promocje cenowe na wybrane produkty (tzn. stara cena X zł -> promocja 10% -> nowa cena Y zł);
   b. Do wybranych pięciu produktów należy zdefiniować co najmniej dwa warianty produktowe tzn. dla obuwia może to być rozmiar 37, 38, 39, dla ubrań może to być rozmiar S, M, L; dla kursów językowych może to być zakres kursu: podstawowy, średnio-zaawansowany itd.
9. **(5 pkt)** Wykonano eksport ustawień sklepu i zapisano jego rezultaty w repozytorium Git projektu. Eksport ustawień powinien umożliwić przywrócenie ustawień sklepu w przypadku awarii.

* - minimalne wymagane do spełnienia, aby uzyskać zaliczenie projektu na ocenę 3,0
** - wymagania będące alternatywą tzn. albo zespół realizuje wymaganie nr 5 albo zespół realizuje wymagania nr 6

Łącznie w części niezbędnej do zaliczenia etapu mogą Państwo uzyskać **20 pkt**
Za realizację zadań dodatkowych mogą Państwo uzyskać również **20 pkt**

Procent pkt	Suma pkt	Ocena
100%	40	5
90%	36	4,5
75%	30	4
65%	26	3,5
50%	20	3

Wymagania na 2. etap (14-15. godzina zajęć projektowych)

1. (5 pkt) * Sklep został wdrożony na klastrze i posiada funkcjonalność określoną co najmniej jako minimalne wymagania określone w etapie I. Sklep korzysta z bazy danych utworzonej na wspólnym serwerze bazodanowym dostępnym w klastrze.
2. (5 pkt) * Zespół utworzył niezbędne pliki Dockerfile oraz plik kompozycji docker-compose.yml, który umożliwia automatyczne wdrożenie sklepu na klastrze studenckim. Kompozycja wykorzystuje jedynie wcześniej zbudowane obrazy, opublikowane w publicznym rejestrze. Utworzona kompozycja została wykorzystana w celu wdrożenia sklepu na klastrze studenckim. Zespół uruchomił dokładnie jeden stack na klastrze i spełnia on wymagania w zakresie nazewnictwa i numerów portów.
3. (5 pkt) * Sklep zintegrowano z usługą Google Analytics. W portalu GA utworzono nową witrynę do monitorowania. Podczas prezentacji GA powinno zawierać dane dotyczące historii odwiedzin sklepu, a także informacje o wartości złożonych zamówień. Informacje powinny rejestrowane co najmniej od 48 godzin.
4. (5 pkt) * Skrypt wykonujący testy automatyczne działa prawidłowo na sklepie wdrożonym w klastrze.
5. (5 pkt) Zespół przygotował kompozycję i dodatkowe skrypty w taki sposób, aby zawartość bazy danych była automatycznie inicjowana podczas uruchamiania kompozycji. Po zainicjowaniu bazy danych, sklep jest gotowy do działania i spełnia minimalne wymagania funkcjonalne wskazane w etapie 1.
6. (3 pkt) W kompozycji zdefiniowano ograniczenia górne na zasoby sprzętowe dla poszczególnych usług tj. vCore oraz ilość dostępnej pamięci RAM per usługa.
7. (2 pkt) Sklep ma włączony cache podczas prezentacji;
8. (5 pkt) Zespół przygotował w ramach źródeł projektu plik konfiguracyjny pipeline wyzwalany automatycznie po zdarzeniu push do gałęzi master/main. W wyniku działania pipeline tworzony jest obraz sklepu.
9. (5 pkt) W Google Analytics zdefiniowano dwa mierzalne cele: typu „destination” oraz typu „event”.
    a. Dla celu typu „destination” należy zdefiniować osiągnięcie przez użytkownika strony po naciśnięciu przycisku rejestracji.
    b. Dla celu typu „event” należy zdefiniować własny typ eventu (i zarejestrować go w GA), który będzie emitowany przez sklep podczas korzystania z baneru lub podczas dodawania przez użytkownika do koszyka produktów w promocji. Po stronie sklepu zaimplementowano emitowanie zdefiniowanych eventów.
    c. Zespół potrafi zaprezentować w panelu GA historię osiągnięcia obydwu zdefiniowanych celów.

Łącznie w części niezbędnej do zaliczenia etapu mogą Państwo uzyskać **20 pkt**
Za realizację zadań dodatkowych mogą Państwo uzyskać również **20 pkt**

Procent pkt	Suma pkt	Ocena
100%	40	5

90%	36	4,5
75%	30	4
65%	26	3,5
50%	20	3

Proces oddawania projektu

1. Rozwiązania dla etapu 1. i etapu 2. można przedstawić do oceny na zajęciach projektowych, nie później niż w terminie wskazanym przez prowadzącego.
2. Wszyscy członkowie zespołów zamierzający przedstawić swoje rozwiązanie do oceny proszeni są o przybycie na początku zajęć projektowych.
3. Każdy członek zespołu powinien być zaznajomiony z opracowanymi przez zespół rozwiązaniami i powinien być gotowy do prezentacji wszystkich aspektów projektu.
4. Na początku zajęć projektowych każdy zespół będzie miał 15 minut na przygotowanie komputera/komputerów i zespołu do prezentacji rozwiązania. Obejmuje to m.in.:
    a. Maszyna wirtualna (w etapach które jej wymagają) powinna być włączona wraz z niezbędnymi usługami;
    b. Kompozycja (w etapach które jej wymagają) powinna być włączona wraz z niezbędnymi usługami;
    c. W przeglądarce internetowej otworzono strony:
        i. repozytorium kodu,
        ii. stronę startową utworzonego sklepu,
        iii. stronę GA z informacjami dla sklepu (jeśli wymagana/zrealizowana podczas etapu),
        iv. stronę ze sklepem źródłowym.
    d. W konsoli należy przygotować kartę/okno:
        i. oczekujące na uruchomienia skryptu do automatycznego testowania;
        ii. oczekujące na wydawanie poleceń na klastrze (etap 2.);
    e. Edytor prezentujący kod źródłowy:
        i. scrappera;
        ii. skryptu inicjującego stan bazy danych;
        iii. pliki Dockerfile i docker-compose.yml;
        iv. skryptu do testów automatycznych
5. Zespół na polecenie prowadzącego uruchomi skrypt do automatycznego testowania sklepu (Selenium). Skrypt nie może zakończyć się błędem, ani generować błędów widocznych dla klienta sklepu (zespół ma 1 podejście i drugie warunkowe jeśli prowadząca/prowadzący wyrazi na to zgodę).
6. Zespół na polecenie prowadzącego wykona dodatkowe czynności sprawdzające w obrębie utworzonego sklepu, a także wykona zlecone niewielkie modyfikacje.
7. Po oddaniu etapu 2. i uzyskaniu pozytywnej oceny, zespół zobowiązany jest wyczyścić klaster studencki ze swoich kompozycji, bazy danych i plików roboczych.

Ocenianie

1. Projekt dzieli się na dwa etapy.
2. Zespół otrzymuje dwie oceny, po jednej za każdy etap.
3. Końcowa ocena z projektu to średnia arytmetyczna ocen z dwóch etapów (zaokrąglana w dół do najbliższej oceny).
4. Ocena 5,0 jest przyznawana za spełnienie wszystkich wymagań określonych w etapie,
5. Ocena 3,0 jest przyznawana wtedy, gdy rozwiązanie spełnia minimalne wymagań, zaznaczonych symbolem *,

6. Pozostałe oceny są do uzyskania według skali liniowej. Nie ma możliwości uzyskania dodatkowych punktów, jeśli wymagania oznaczone symbolem * nie zostały spełnione.

Terminy realizacji

Dzienne:
I etap: prezentacja podczas zajęć projektowych obejmujących 9-10. godzinę zajęć dla zespołu
II etap: prezentacja podczas zajęć projektowych obejmujących 14-15. godzinę zajęć dla zespołu

Dokładne terminy zależą od planu zajęć i ewentualnych przerw świątecznych.

Wskazówki

Opis studenckiego klastra katedralnego oraz stosowanie dockera zostały opisane w instrukcji: Wskazówki techniczne

Proszę upewnić się, że działa Państwu połączenie do wydziałowego serwera VPN: http://starter.eti.pg.gda.pl/

Jak testować własny sklep wdrożony na klastrze za pomocą przeglądarki internetowej zainstalowanej na swoim komputerze?

W instrukcji do klastra znajdą Państwo wiele pomocnych materiałów m.in. jak zestawić tunel SSH, przez który będą Państwo mieli możliwość komunikacji z aplikacją wdrożoną na klastrze. Proszę stosować tę instrukcję jeśli posiadają Państwo tunel VPN do wydziału ETI.

Niniejsza procedura dotyczy nawiązywania połączenia w przypadku braku VPN wydziałowego. Wymaga posiadania konta na serwerze kask.eti.pg.gda.pl. Zakładamy tutaj, że Państwa usługa nasłuchuje w klastrze na porcie **PORT1**. Obydwa polecenia należy wydać w osobnych terminalach, których nie można potem zamykać do czasu zakończenia testów usługi. Polecenia należy wydać w kolejności w jakiej zostały tutaj określone:

ssh -L PORT1:172.20.83.101:22 kask.eti.pg.gda.pl

ssh -L PORT2:student-swarm01.maas:PORT1 hdoop@localhost

Po zestawieniu połączenia rekomendowane jest, aby w każdym terminalu uruchomić polecenie **top.** Dzięki temu terminal automatycznie nie rozłączy się po określonym czasie bezczynności.

Po wydaniu powyższych poleceń możliwe będzie komunikowanie się z usługą po stronie klastra. W tym celu proszę uruchomić przeglądarkę internetową i wpisać następujący adres:

```
localhost:PORT2
```