

Programowanie w Internecie

projekt

2017/2018

K.M. Ocetkiewicz

Zasady ogólne

Za projekt można otrzymać maksymalnie 100 pkt. Projekt podzielony jest na trzy etapy. Na wyniki z poszczególnych etapów wpływ będą miały:

- zgodność ze specyfikacją projektu,
- spełnienie wymagań ogólnych,
- odpowiedź na zadane przez prowadzącego pytania – mogą dotyczyć kodu, tematyki etapu lub wymagać modyfikacji prezentowanego kodu,
- oddanie zadania w terminie – implementacja rozwiązania musi być przekazana prowadzącemu do momentu zakończenia zajęć w dniu wyznaczonym jako koniec etapu,
- samodzielność rozwiązania – oceniany będzie tylko kod napisany samodzielnie, którego działanie autor jest w stanie wyjaśnić (nie należy liczyć na punkty za kod którego się nie rozumie),
- nietrywialność rozwiązania – wyżej punktowane będą fragmenty kodu które rzeczywiście realizują funkcjonalność serwisu niż te, które służą „zaliczeniu” danego wymagania; w razie braku nietrywialnego zastosowania wymagania ogólnego w projekcie należy ustalić z prowadzącym sposób rozwiązania (np. zmiana czy dostosowanie wymagania do projektu).

Udzielenie odpowiedzi następuje wraz z oddaniem implementacji. Istnieje możliwość odpowiedzi w terminie późniejszym, o ile termin ten został uzgodniony przed oddaniem implementacji, nie przekracza terminu ostatnich zajęć i źródła zostały dostarczone w wymaganym terminie. Oddanie etapu w terminie wcześniejszym jest zawsze dozwolone. Oceny poszczególnych etapów są niezależne.

Etap 0: specyfikacja projektu

Należy wybrać tematykę implementowanego serwisu i zgłosić ją prowadzącemu do akceptacji (e-mailem lub na zajęciach). Zgłoszenie powinno zawierać temat oraz przewidywany zakres funkcjonalności (o poziomie szczegółowości takim jak w przykładowych tematach pod koniec tego dokumentu). Projekt powinien posiadać nietrywialną część kliencką i serwerową oraz korzystać z bazy danych. Oddanie specyfikacji po terminie skutkuje utratą punktów. Brak akceptacji specyfikacji uniemożliwia oddanie kolejnych części projektu.

Brak odpowiedzi w przeciągu tygodnia na e-mail ze zgłoszeniem tematu proszę uznać za akceptację.

Uwaga: tematy „Sklep z ...” nie będą akceptowane.

termin: 18 XI 2017 (liczy się termin pierwszego przesłania specyfikacji)

liczba punktów: 5

Etap 1: prezentacja postępu prac

Etap ten polega na przedstawieniu postępu prac nad projektem. Projekt powinien być ukończony w ok. 50% (np. zaimplementowana połowa całości lub cała część kliencka lub serwerowa). Etap polega na przedstawieniu kodu i odpowiedzi na pytania. Oddanie tego etapu 18 XI 2017 daje 20% punktów więcej.

termin: 9 XII 2017

liczba punktów: 48

Etap 2: zakończenie projektu

Etap polega na przedstawieniu ostatecznej wersji kodu i odpowiedzi na pytania dotyczące tego kodu. Ostateczne oddanie projektu nie później niż 13 I 2018 jest premiowane dodatkowymi sześcioma punktami.

termin: 27 I 2018

liczba punktów: 47

Uwagi

28 I 2018 to ostateczny i nieprzekraczalny termin oddawania i odpowiedzi z projektów.

Każdy etap ma maksymalną liczbę punktów, której nie można przekroczyć. Oznacza to, że zignorowanie etapu 1 (dostając za niego 0 punktów) obniży maksymalną liczbę punktów do zdobycia do 52 – nawet, jeżeli projekt zrealizowany będzie w 100% podczas etapu 2. Z drugiej strony, w terminie etapu 1 można otrzymać 95 punktów, jeżeli odda się oba etapy równocześnie (o ile będzie dość czasu na oddanie wszystkiego).

Korzystanie z zewnętrznych modułów czy bibliotek (np. jQuery) nie jest zabronione, ale części projektu je wykorzystujące nie będą oceniane.

Niezależnie od sposobu utworzenia bazy danych, do źródeł projektu należy dołączyć kod SQL tworzący bazę danych i wypełniający ją danymi niezbędnymi do pełnej prezentacji projektu. W przypadku korzystania z narzędzi typu ORM czy dostępu do bazy danych z poziomu frameworka, bazę danych należy utworzyć samodzielnie i korzystać z niej, jakby nie było możliwości modyfikacji jej schematu z poziomu frameworka. Np.:

źle	dobrze
<pre>class User { int id; string login, password; } framework.CreateDatabaseFor(typeof(User));</pre>	<pre>sql: create table users (id int, nam text, pwd text); kod: [DatabaseTable("users")] class User { [DatabaseField("id")] int id; [DatabaseField("nam")] string login; [DatabaseField("pwd")] string pass; } framework.ConnectToDatabase("DNS=db");</pre>

Wymagania ogólne

Niezależnie od tematyki, każdy projekt powinien spełniać poniższe wymagania.

Część serwerowa (47 pkt.):

- należy wykorzystać bazę danych (insert, select, delete, update, złożone zapytanie: sortowanie + złożony warunek, złączenia itp.) (10 pkt.),
- należy odpowiednio zabezpieczyć zawartość bazy danych i dostęp do niej (niewrażliwość na SQL injection, kodowanie i solenie haseł) oraz przeanalizować wydajność zapytań (odpowiednie indeksy) (5 pkt.),
- należy wykorzystać mechanizm sesji (3 pkt.),
- należy świadomie skorzystać z ciasteczek (np. do przechowywania wyboru języka; implementacja sesji się nie liczy) (3 pkt.),
- serwer powinien dynamicznie generować co najmniej 3 różne typy dokumentów (np. HTML, JSON, pliki tekstowe) (5 pkt.),
- obsługa przesyłania plików na serwer z kontrolą typu i rozmiaru (5 pkt.),
- mechanizm przesyłania plików powinien być odporny na ataki (directory traversal) (3 pkt.),
- przynajmniej 5 komunikatów w serwisie powinno mieć co najmniej dwie wersje językowe; język należy pobierać z nagłówek zapytania, z możliwością wymuszenia innego (np. przez parametry zapytania); rozszerzanie o kolejne języki powinno być trywialne (podczas oddawania proszę spodziewać się prosby o dodanie komunikatów w kolejnym języku) (6 pkt.),
- co najmniej dwie wersje kolorystyczne strony z możliwością wyboru przez użytkownika (wybór powoduje dołączenie odpowiedniego arkusza stylów) (4 pkt.),
- treści prezentowane użytkownikom powinny być bezpieczne (odporność na cross-site scripting) (3 pkt.).

Część kliencka (48 pkt.):

- formatowanie wykorzystujące CSS (5 pkt.),
- układ strony zdefiniowany poprzez CSS (3 pkt.),
- wykorzystanie tagów semantycznych (2 pkt.),
- wykorzystanie języka JavaScript; kod powinien implementować nietrywialne funkcje serwisu i mieć znaczną objętość (ok. 100 linii) (8 pkt.),
- wykorzystanie JavaScript do modyfikacji drzewa DOM (5 pkt.),
- wykorzystanie HTML5 (np. element canvas, FileAPI, local storage, web workers, WebGL itp.) (8 pkt.),
- wykorzystanie komunikacji asynchronicznej z serwerem (AJAX) (5 pkt.),
- cały kod JavaScript w trybie strict (2 pkt.),
- strona powinna zostać zwalidowana pod kątem zgodności ze standardem HTML / XHTML (<http://validator.w3.org>) (4 pkt.),
- strona powinna zostać zwalidowana pod kątem zgodności ze standardem CSS (<http://jigsaw.w3.org/css-validator/>) (2 pkt.),
- strona nie powinna istotnie różnić się wyglądem w różnych przeglądarkach (4 pkt.).

Zaawansowane wykorzystanie niektórych narzędzi może być premiowane dodatkowymi punktami (ok. 5 pkt.). Przykłady:

- zaawansowany CSS: przewijalna galeria zdjęć z możliwością ich powiększania nie korzystająca z JavaScriptu,
- zaawansowany CSS: rozwijalne, „klikalne” menu służące do nawigacji po stronie nie korzystające z JavaScriptu,
- zaawansowana modyfikacja DOM: tworzenie okien dialogowych na stronie z wykorzystaniem samodzielnie interpretowanych szablonów,
- zaawansowany JavaScript: implementacja sztucznej inteligencji w grze.

Uwagi dotyczące łączenia projektów z różnych przedmiotów

Stworzenie jednego projektu na dwa różne przedmioty (np. Programowanie w Internecie i Systemy Informacji Przestrzennej) jest możliwe. Należy jednak rozdzielić funkcjonalności wymagane na różnych przedmiotach. Na przykład, umieszczona na stronie i oprogramowana w JavaScriptcie mapa (na SIP) nie zaliczy punktu „wykorzystanie JavaScriptu” z PwI (nie jest to samodzielnie napisany kod lecz skorzystanie z gotowego narzędzia czy modułu).

Przykładowe tematy projektów

Internetowa gra dwuosobowa (szachy/warcaby/statki/inna)

Serwis umożliwiający przeprowadzenie graczom rozgrywki w wybraną grę dwuosobową (nie jest wymagana implementacja „sztucznej inteligencji”):

- rejestrowanie użytkowników, identyfikowanych przez login i hasło,
- stworzenie nowej gry,
- przeglądanie listy gier,
- dołączanie do gry,
- przeprowadzenie rozgrywki,
- tryb obserwatora: możliwość dołączenia do gry i obserwowania jej przebiegu,
- pamiętanie historii rozgrywki – użytkownik może prześledzić przebieg każdej rozgrywki, w której brał udział,
- rozgrywka odbywa się bez przeładowania strony (wołania asynchroniczne); dotyczy to także trybu obserwatora oraz odtwarzania minionej rozgrywki.

Edytor plików tekstowych online

Każdy zarejestrowany użytkownik może przechowywać w serwisie kilka plików tekstowych, które może przesłać na serwer, pobrać z serwera i edytować na stronie serwisu:

- rejestrowanie użytkowników, identyfikowanych przez login i hasło,
- możliwość tworzenia, usuwania i edycji plików,
- możliwość pobrania wybranego przez użytkownika pliku na dysk komputera,
- możliwość dodania do listy plików nowego, przesłanego przez użytkownika,
- edycja pliku powinna zapewniać:
 - obsługę podstawowych klawiszy edycyjnych (strzałki, Page Up, Page Down, Home, End, Delete, Backspace, Enter),
 - możliwość wyboru rozmiaru tabulacji (1, 2, 3, 4, 8 spacji itp.),
 - możliwość wyboru kroju liter, ich wielkości oraz interlinii,

- kolorowanie składni w podstawowym zakresie (wyróżnianie słów kluczowych i komentarzy),
- zapamiętanie zmodyfikowanego pliku pod starą lub nową nazwą.

Wirtualne płótno

Serwis pozwalający na grupową edycję obrazu:

- rejestrowanie użytkowników, identyfikowanych przez login i hasło,
- możliwość utworzenia nowego obrazka,
- możliwość dołączenia do edycji wybranego obrazka,
- obrazek jest trwały – jest pamiętany nawet, gdy wszyscy zakończą jego edycję; do edycji obrazka można powtórnie dołączyć,
- możliwość zapisania edytowanego obrazka na dysk komputera,
- rysowanie prostych figur geometrycznych: punkt, linia, strzałka, okrąg, elipsa, prostokąt, tekst,
- możliwość wyboru koloru i szerokości linii oraz koloru wypełnienia (wybrany kolor lub brak).

Sklep internetowy

Strona sklepu handlującego przedmiotami z wybranej dziedziny:

- możliwość przeglądania oferty, podzielonej na działy,
- każdy produkt posiada nazwę, opis, cenę, oraz zdjęcie,
- wyszukiwanie produktów według nazwy, działu, opisu,
- możliwość sortowania wyniku wyszukiwania alfabetycznie według nazwy oraz według ceny (rosnąco i malejąco),
- wynik wyszukiwania podzielony na strony po 5, 10, 20, 50 produktów (do wyboru przez użytkownika),
- możliwość wyboru formatu wyniku wyszukiwania (lista nazw i cen produktów, lista z nazwą ceną i zdjęciem lub zdjęcia rozłożone w kolumnach i wierszach),
- sklep posiada bazę klientów, identyfikowanych przez login i hasło,
- zalogowany użytkownik może dodawać produkty do koszyka oraz składać zamówienie,
- panel administracyjny umożliwiający dodawanie, usuwanie i edycję produktów (zmiana zdjęcia, ceny, opisu, przypisania do działu), działów (tworzenie, zmiana nazwy, usuwanie) oraz zamówień (przeglądanie, zmiana stanu: przyjęte/realizowane/wysłane).

Tematy „Sklep z ...” i pochodne nie będą akceptowane aby zapewnić równość szans.

Prosty interfejs do zarządzania zawartością bazy danych przez WWW

Strona umożliwiająca zarządzanie zawartością wybranej bazy danych przez WWW:

- możliwość wprowadzenia przez użytkownika nazwy bazy danych do zarządzania i hasła
- możliwość tworzenia tabel z dowolną ilością pól o wybranych nazwach, typach i rozmiarach (zbiór typów powinien obejmować do najmniej int, varchar, bool) i wskazania klucza głównego
- możliwość edycji zawartości bazy (dodawanie rekordu do tabeli, usuwanie rekordu, modyfikacja zawartości)

- wprowadzanie/edycja pola powinna być ograniczona do danego typu (bool jako checkbox lub wybór spośród TAK/NIE, int jako pole tekstowe z możliwością wpisania tylko cyfr, napis jako pole tekstowe)
- edycja pojedynczych pól powinna odbywać się bez przeładowania strony (zapytania asynchroniczne)
- możliwość wykonania dowolnego, wprowadzonego przez użytkownika, zapytania i przedstawienie rezultatu użytkownikowi, podzielonego na strony po 10, 20 lub 50 rezultatów (do wyboru przez użytkownika)
- komunikaty błędów bazy danych powinny być prezentowane użytkownikowi
- w sesji powinna być pamiętana historia wszystkich wykonanych operacji; użytkownik powinien mieć możliwość powtórzenia wykonania każdej z nich (poprzez interfejs z wcześniejszego punktu)
- strona powinna mieć co najmniej dwie wersje językowe (polska/angielska); wybór wersji powinien być uzależniony od ustawień przeglądarki.

Narzędzie do zbierania statystyk dla witryn internetowych

Narzędzie umożliwiające gromadzenie statystyk dotyczących odwiedzin stron internetowych:

- gromadzenie statystyk dla strony powinno odbywać się poprzez osadzenie w niej obrazka
- projekt powinien gromadzić następujące dane z nagłówków HTTP (indywidualnie dla każdej witryny): adres IP, rodzaj i wersję przeglądarki, system operacyjny, preferowany język, kraj (za pomocą np. <http://www.hostip.info/use.html>), data i czas dostępu
- narzędzie powinno udostępniać interfejs do przeglądania statystyk:
 - prezentowanie zapytań pogrupowanych według (tabela uporządkowana według ilości zapytań): kraju, języka, systemu operacyjnego, przeglądarki
 - prezentowanie danych na wykresach (wykorzystanie znacznika canvas) z czasem na poziomej osi:
 - historia wszystkich dostępu
 - historia dostępu z podziałem na język
 - historia dostępu z podziałem na system operacyjny
 - historia dostępu z podziałem na przeglądarkę
- interfejs powinien posiadać podstronę prezentującą zebrane od użytkownika dane (np. podstrona [info.html](#) prezentująca kraj, język, system operacyjny, nazwę przeglądarki adres ip. itp. klienta, który załadował tę stronę).

Wirtualny dysk

Portal umożliwiający zarejestrowanym użytkownikom przechowywanie swoich danych:

- narzędzie powinno umożliwiać wysyłanie, pobieranie i przeglądanie plików
- z narzędzia korzystają użytkownicy, którzy identyfikują się podając login i hasło
- użytkownicy widzą tylko własne pliki
- możliwość rejestracji nowych użytkowników
- pliki mogą być podzielone na katalogi (katalog może zawierać podkatalogi)
- przeglądanie powinno prezentować, dla bieżącego katalogu: nazwę pliku/katalogu, datę wysłania na serwer, typ pliku, rozmiar
- tabela prezentująca listę plików powinna umożliwiać sortowanie (kliknięcie na nagłówku "nazwa" powinno sortować według nazwy)

- nawigacja po katalogach i sortowanie nie powinno przeładowywać całej strony (zapytania asynchroniczne)
- kliknięcie na nazwę pliku powinno (rozpoznawanie po rozszerzeniu):
 - prezentować zawartość plików tekstowych i źródłowych (txt, cpp, h, js, html, itp.)
 - wyświetlać obrazki (jpg, png, gif, itp.)
 - kliknięcie na pozostałych plikach powinno umożliwiać ich pobranie
- kliknięcie na nazwie katalogu powinno przenosić użytkownika do tego katalogu
- możliwość wysyłania plików przekraczających maksymalny rozmiar zapytania w ustawieniach PHP (zob. FileAPI w JavaScript)
- wysyłanie powinno wykorzystywać pasek postępu, prezentujący jaki fragment pliku został już wysłany.

Arkusz kalkulacyjny online

Strona umożliwiająca kilku osobom pracę nad jednym, współdzielonym arkuszem kalkulacyjnym:

- rejestrowanie użytkowników, identyfikowanych przez login i hasło,
- możliwość utworzenia nowego arkusza,
- możliwość dołączenia do edycji wybranego arkusza,
- arkusz jest trwały – jest pamiętany nawet, gdy wszyscy zakończą jego edycję; do edycji arkusza można powtórnie dołączyć,
- obsługa podstawowych działań matematycznych w arkuszu,
- obsługa odwołań do innych komórek (np. komórka zawierająca sumę dwóch poprzednich),
- obsługa prostych funkcji statystycznych (suma, średnia, iloczyn wielu komórek),
- dostęp tylko do odczytu do arkuszy dla niezarejestrowanych użytkowników.

Komunikator internetowy

Strona umożliwiająca użytkownikom porozumiewanie się w czasie rzeczywistym:

- rejestrowanie użytkowników, identyfikowanych przez login i hasło,
- lista znajomych, z którymi możemy się komunikować (dodawanie, usuwanie),
- możliwość ustawiania awatarów,
- ustawianie statusu użytkownika, widocznego dla znajomych (dostępny, zajęty itp.),
- komunikacja w czasie rzeczywistym (tekst, emotikony, obrazki),
- informacja o próbie nawiązania kontaktu przez znajomego,
- przesyłanie plików między użytkownikami,
- pobranie całej treści rozmowy,
- szyfrowanie komunikacji i przesyłanych plików w celu zachowania prywatności.