

PROJEKT – ETAP 2

Wytyczne implementacyjne

W ramach drugiego etapu projektu z przedmiotu Wytwarzanie Aplikacji Internetowych należy zaimplementować aplikację galerii zdjęć. Jako tematykę do realizacji projektu należy wykorzystać stronę przygotowaną w ramach pierwszego etapu. Tworzona w tym etapie galeria zdjęć powinna wpisywać się w tematykę strony, np. do strony o filmach – galeria aktorów. Można wykorzystać układ i formatowanie opracowane w etapie 1.

Strona serwerowa powinna zostać zaimplementowana w języku PHP, bez wykorzystania zewnętrznych frameworków. Galeria musi pozwalać na dodawanie nowych plików graficznych i przeglądanie tych, które zostały uprzednio przesłane na serwer. Aplikacja powinna ponadto umożliwiać prawidłową rejestrację użytkowników, logowanie i wylogowanie. Informacje prezentowane na stronie powinny się różnić w zależności od tego, czy użytkownik jest zalogowany czy nie – zgodnie ze szczegółowymi wytycznymi w dalszej części.

W celu realizacji projektu należy zaimplementować poniższe zadania cząstkowe. Zostały one przypisane do 3 kategorii, reprezentujących poziom ich trudności (kat. I – proste, kat. II – o średnim poziomie trudności i kat. III – trudniejsze). Wykonanie zadań z wyższej kategorii nie zwalnia z realizacji zadań łatwiejszych. Nie ma jednak obowiązku implementowania wymagań w kolejności podanej poniżej. Aby przystąpić do prezentacji projektu konieczne jest zaimplementowanie wszystkich zadań, oznaczonych jako „wymagane”.

W celu uzyskania maksymalnej liczby punktów (40) z tej części projektu należy wykonać wszystkie poniższe zadania (za łącznie 31 pkt), uzyskać współczynnik 1,0 za architekturę (szczegóły poniżej) oraz odpowiedzieć na pytania do każdej z trzech kategorii (w każdej z nich do zdobycia po 3 pkt). Jeśli z danej kategorii nie zostało wykonane ani jedno zadanie, prowadzący nie zadaje pytań z nią związanych i nie przysługują punkty za odpowiedź w tej kategorii.

Punkty uzyskane w tym etapie uzależnione są od architektury przygotowanej aplikacji. Zagadnienia architektury aplikacji internetowych i modelowania przepływu sterowania będą omówione na wykładzie wraz z przykładami. Student w ramach odpowiedzi ma obowiązek wykazać się znajomością/zrozumieniem zastosowanej architektury. W ramach projektu wyróżnia się następujące kryteria oceny architektury i wynikające z nich współczynniki:

Kryteria	Współczynnik
Pomieszczone elementy logiki biznesowej i interfejsu aplikacji, niezachowana właściwa kolejność przetwarzania.	0,6
Zachowana właściwa kolejność przetwarzania: obsługa parametrów zapytania HTTP, logika biznesowa, ustawienie parametrów odpowiedzi HTTP, prezentacja widoku. Wydzielona logika biznesowa (np. w postaci funkcji biznesowych).	0,8
Funkcyjna implementacja wzorca MVC: front-controller, routing, kontrolery, model, widoki + wydzielona logika biznesowa (np. w postaci funkcji biznesowych)	1,0
<i>Dla chętnych na dodatkowe punkty</i> Obiektowa implementacja wzorca MVC + wydzielona logika biznesowa	1,2

Końcowa punktacja za ten etap projektu jest obliczana ze wzoru:

$$P_{\text{total}} = P_{\text{wymagania}} * w + P_{\text{odpowiedzi}}$$

gdzie:

P_{total} – końcowa liczba punktów zdobytych przez studenta,

$P_{\text{wymagania}}$ – liczba punktów za realizację wymagań funkcjonalnych (tabela poniżej),

w – współczynnik za jakość architektury aplikacji,

$P_{\text{odpowiedzi}}$ – liczba punktów za odpowiedzi udzielone w czasie prezentacji projektu.

Szczegółowe wymagania funkcjonalne znajdują się w tabeli poniżej:

Zadanie	Punkty
Kategoria I	
A) Przesyłanie plików na serwer	2 (wymagane)
<p>Aby dostarczyć zdjęcia do galerii, należy zaimplementować funkcję przesyłania plików na serwer (ang. <i>upload</i>). Dozwolone powinny być wyłącznie pliki w formacie PNG lub JPG. Przesyłane pliki nie mogą przekraczać rozmiaru 1 MB. Jeśli użytkownik spróbuje wysłać zdjęcie o większym rozmiarze lub w innym formacie, powinien otrzymać stosowny komunikat (osobno dla formatu pliku, jego wielkości lub oba komunikaty jednocześnie).</p> <p>W katalogu, skonfigurowanym jako DocumentRoot aplikacji, należy dodać podkatalog <code>images</code> i umieszczać w nim przesłane na serwer zdjęcia [zwróć uwagę na uprawnienia do tego katalogu]. Dzięki temu serwer WWW będzie mógł później zaserwować zdjęcia przeglądarce, w celu ich wyświetlenia na stronie.</p>	
B) Znak wodny i miniatura	4
<p>Formularz przesyłania plików należy uzupełnić o pole tekstowe zatytułowane „znak wodny”, którego wypełnienie jest obowiązkowe.</p> <p>Po przesłaniu zdjęcia, po stronie serwera należy dokonać dwóch przekształceń otrzymanego pliku:</p> <ul style="list-style-type: none">• nałożyć na oryginalne zdjęcie tekstowy znak wodny (treść znaku pobrana z formularza),• wygenerować miniaturkę o rozmiarze 200x125 pikseli. <p>Wykorzystać należy bibliotekę PHP GD, która została zainstalowana w maszynie wirtualnej, w czasie konfiguracji środowiska pracy.</p> <p>Po zaimplementowaniu powyższych operacji, na serwerze powinny znajdować się 3 pliki graficzne dla każdego przesłanego zdjęcia: 1) oryginalne zdjęcie, 2) zdjęcie ze znakiem wodnym oraz 3) miniaturka.</p>	
C) Galeria przesłanych zdjęć	4 (wymagane)
<p>Należy przygotować kod PHP, pozwalający na wyświetlenie zbiorczej galerii wszystkich zdjęć, wysłanych na serwer. Galeria powinna prezentować miniaturki zdjęć (wygenerowane w ramach zadania I.B powyżej) na zasadzie stronicowania (<i>paging</i>) po xx sztuk na raz [wartość xx według preferencji twórcy]. Jeśli zadanie I.B nie zostało wykonane, należy wykorzystać oryginalne zdjęcia, ale zaprezentowane w pomniejszeniu przy użyciu stylów CSS.</p> <p>Kliknięcie na miniaturę zdjęcia powinno kierować do zdjęcia ze znakiem wodnym w pełnym rozmiarze (jeśli zadanie I.B nie został zaimplementowany, należy wyświetlić oryginalne zdjęcie w pełnym rozmiarze).</p> <p>W ramach zadania należy wykorzystać przynajmniej dwie spośród następujących dyrektyw języka</p>	

PHP: include, include_once, require, require_once. Zastosowania muszą być zgodne z przeznaczeniem wybranych dyrektyw.	
Kategoria II	
A) Wykorzystanie bazy danych	3
Formularz wysyłania plików należy uzupełnić o pola tekstowe „tytuł” oraz „autor”. Po otrzymaniu poprawnego pliku (format i rozmiar), dane o zdjęciu należy zapisać w bazie MongoDB. Informacje o tytułach i autorach poszczególnych zdjęć należy wyświetlać w galerii miniatur.	
B) Rejestracja i logowanie użytkowników	5
<p>Należy przygotować formularz rejestracji użytkowników, zawierający pola: „adres e-mail”, „login”, „hasło”, „powtórz hasło” (wszystkie pola obowiązkowe). Po zweryfikowaniu danych, otrzymanych z formularza (m.in. należy sprawdzić, czy wpisany login nie został już zajęty), należy zapisać nowego użytkownika w bazie danych MongoDB. Należy pamiętać o przekształceniu hasła do postaci skrótu, przy użyciu bezpiecznej funkcji hashującej.</p> <p>Należy przygotować formularz logowania, zawierający pola „login” i „hasło”. Do uwierzytelniania należy wykorzystać bazę zarejestrowanych użytkowników aplikacji. Jeśli podany login i hasło pasują do jednego z istniejących w bazie danych użytkowników, należy zapisać w sesji jego identyfikator i wyświetlić komunikat o pomyślnym logowaniu. W przeciwnym razie należy ponownie wyświetlić formularz logowania wraz z komunikatem o niepowodzeniu.</p> <p>Dla użytkownika zalogowanego formularz logowania nie powinien być widoczny.</p> <p>Dla użytkownika zalogowanego dostępna możliwość wylogowania;</p>	
C) Mechanizm sesji	4
<p>Należy zmodyfikować galerię miniatur tak, aby przy każdym zdjęciu widoczne było pole typu <i>checkbox</i>, identyfikujące dane zdjęcie (można wykorzystać wartości pola <i>_id</i> z bazy danych). Należy również dodać przycisk „Zapamiętaj wybrane”. Gdy użytkownik zaznaczy zdjęcia i kliknie przycisk, należy zapamiętać jego wybór w sesji. Gdy użytkownik po raz kolejny wejdzie na stronę galerii miniatur, zapamiętane wcześniej zdjęcia powinny mieć zaznaczone przypisane im pola typu <i>checkbox</i>.</p> <p>Należy dodać kolejną podstronę w aplikacji, prezentującą wyłącznie zdjęcia, które zostały wcześniej wybrane przez użytkownika. Przy wszystkich zdjęciach na tej podstronie ponownie należy wyświetlić pola typu <i>checkbox</i>, które pozwolą na usunięcie zaznaczonych zdjęć ze zbioru zapamiętanych w sesji (po kliknięciu przycisku „Usuń zaznaczone z zapamiętanych”).</p> <p><i>Funkcjonalność jest analogiczna do dodawania/usuwania produktów do/z koszyka zakupowego.</i></p>	
Kategoria III	
A) Rozróżnianie użytkowników	5
<p>Formularz przesyłania zdjęć należy zmodyfikować tak, aby oferował dodatkowe funkcje zalogowanym użytkownikom. Gdy użytkownik jest zalogowany, pole „autor” formularza powinno automatycznie zawierać login użytkownika. Ponadto dla zalogowanych użytkowników należy wyświetlać dodatkowe pole typu <i>radio</i>, pozwalające na przełączanie pomiędzy wartościami „publiczne” oraz „prywatne”.</p> <p>Jeśli użytkownik wybierze opcję „prywatne”, zdjęcie nie powinno być widoczne w galerii miniatur dla innych użytkowników (zarówno zalogowanych jak i anonimowych). Zdjęcie powinno być widoczne w galerii dla użytkownika, który je przesłał (wraz z informacją, że jest ono prywatne).</p> <p>Zdjęcia publiczne należy obsługiwać tak, jak dotychczas. Zdjęcia przesyłane przez niezalogowanych użytkowników należy automatycznie traktować jako publiczne.</p>	
B) AJAX – wyszukiwarka zdjęć	4
Należy zaimplementować wyszukiwarkę zdjęć, wykorzystującą technologię AJAX. Wyszukiwarka powinna być zrealizowana jako osobna podstrona w aplikacji. Po wejściu na tę podstronę,	

widoczne powinno być pole tekstowe, do którego użytkownik może wpisać fragment tytułu zdjęcia, którego poszukuje.

W miarę wpisywania w wyszukiwarkę kolejnych liter przez użytkownika (np. w reakcji na zdarzenie onkeyup) należy wysyłać do serwera kolejne asynchroniczne żądania HTTP (AJAX), które zawierają aktualną wartość pola tekstowego. Serwer wyszukuje zdjęcia, których tytuły zawierają frazę wpisaną przez użytkownika i w odpowiedzi wysyła fragment kodu HTML, prezentujący miniatury dopasowanych zdjęć. Przeglądarka, po otrzymaniu odpowiedzi z serwera, wyświetla dopasowane miniatury użytkownikowi.

Rozliczenie etapu

Termin realizacji: zgodnie z harmonogramem z pliku „Zasady realizacji i zaliczenia projektu.pdf”.

- Przygotuj archiwum z plikami wytworzonymi w ramach etapu (cała struktura projektu PHP - nie obraz dysku całej maszyny - spakowana do pliku nazwanego wg schematu **nazwisko_imie_numer_albumu.zip**. Paczkę prześlij w sekcji projekt ‘Etap2-strona serwera’ zgodnie z harmonogramem.

Prezentacja: Obowiązują takie same zasady prezentacji projektu jak w poprzednim etapie (zarezerwować w arkuszu termin odpowiedzi, przygotować się do prezentacji, zaprezentować).

- Rozliczenie w zarezerwowanym indywidualnie terminie:
 - ➔ Wydrukuj/przerysuj dołączoną na końcu dokumentu tabelkę.
 - ➔ Wpisz w tabeli swoje dane oraz punkty przy zrealizowanych zadaniach. Zadeklarowanie wykonania wymagań, które nie zostały faktycznie zrealizowane lub przedstawienie kodu odbiegającego od kodu wgranego na enauczaniu, skutkuje uzyskaniem zera punktów z całej części projektowej przedmiotu.
 - ➔ Wypełnioną tabelkę zabierz na rozliczenie projektu; zaprezentuj dzieło, odpowiedz na pytania prowadzącego; Podlicz końcowy wynik punktowy w tabeli. Kartkę z tabelką zostaw prowadzącemu.
- Ostatecznie przysługująca liczba punktów (z uwzględnieniem bonusu lub kary) jest obliczana zgodnie z zasadami realizacji i zaliczania projektu (dostępne na stronie przedmiotu w sekcji *Projekt*) i odnotowywana na platformie enauczanie.

Przed prezentacją należy:

- skonfigurować i przetestować witrynę w środowisku produkcyjnym (port 80)
- przesłać do witryny kilka plików pozwalających zademonstrować działanie wykonanego stronicowania w galerii zdjęć;
- przygotować sobie kilka plików, pozwalających na przetestowanie mechanizmów walidacji zdjęć wysyłanych na serwer. Wśród testowych plików powinny znaleźć się:
 - poprawne pliki graficzne, które zostaną zaakceptowane w galerii,
 - przynajmniej jeden plik o poprawnym rozmiarze i niepoprawnym typie (np. plik w formacie PDF),
 - przynajmniej jeden plik o poprawnym typie, ale o zbyt dużym rozmiarze.
- Otworzyć w edytorze pliki źródłowe kodu [o ile został wykonany: upload pliku, funkcje biznesowe, kod Ajax, logowanie/wylogowanie, operacje na „koszyku” zdjęć - zapamiętanie zdjęć w sesji].
- Prowadzący wybiera funkcjonalności, które student prezentuje w ramach rozliczania etapu.

Etap2			
Imie i nazwisko:		
Nr albumu		
Wymaganie	Punkty	Wykonane?	
Kategoria I	pyt I		
A) Przesyłanie plików na serwer – WYMAGANE	2		
B) Znak wodny i miniatura	4		
C) Galeria przesłanych zdjęć – WYMAGANE	4		
Kategoria II	pyt II		
A) Wykorzystanie bazy danych	3		
B) Rejestracja, logowanie i wylog.użytkowników	5		
C) Mechanizm sesji	4		
Kategoria III	pyt III		
A) Rozróżnianie użytkowników	5		
B) AJAX – wyszukiwarka zdjęć	4		
Razem zrealizowane wymagania (Pwymagania):			
Współczynnik za architekturę (w) *:			
PKT za wykonany kod (pkod = <i>pwymagania</i> · w):			
Razem (pkod+ Podpowiedzi)			
ilość rozpoczętych tyg. opóźnienia			
PKT z uwzględnieniem opóźnienia			
Komórki żółte wypełnia student przed prezentacją!		Komórki czerwone wypełnia prowadzący	

Współczynnik za architekturę - opis:	
<i>Pomieszczone elementy logiki biznesowej i interfejsu aplikacji, niezachowana właściwa kolejność przetwarzania.</i>	0,6
<i>Zachowana właściwa kolejność przetwarzania: obsługa parametrów zapytania HTTP, logika biznesowa, ustawienie parametrów odpowiedzi HTTP, prezentacja widoku. Wydzielona logika biznesowa (np. w postaci funkcji biznesowych).</i>	0,8
<i>Funkcyjna implementacja wzorca MVC: front-controller, routing, kontrolery, model, widoki + wydzielona logika biznesowa (np. w postaci funkcji biznesowych)</i>	1
<i>Dla chętnych na dodatkowe punkty Obiektowa implementacja wzorca MVC + wydzielona logika biznesowa</i>	1,2