

$$E_{\text{wyk}} = \left(\sum_{\text{pkt}=1}^{20} \text{pkt} \cdot \text{uzysk.} \right) \cdot 2$$

Zalicza
Ewyk ≥ 25 pkt.

[B_1] WAI 05.02.2025 Imię, nazwisko, indeks:
W pytaniu testowym za oznaczenie złej odpowiedzi przydzielane jest **minus 50%** pkt z pytania (do zera w ramach pyt.)

		Listing pyt:1-5
1	<!DOCTYPE html>	
2	<html lang="en-EN">	
3	<head>	
4	<meta charset="UTF-8">	
5	<title>Example application</title>	
6	<link href="https://fonts.googleapis.com/css2?family=Open+Sans:wght@600&display=swap"	
7	rel="stylesheet">	
8	<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>	
9	<script src="https://code.jquery.com/ui/1.14.1/jquery-ui.js"></script>	
10	<script>	
11	\$(function() {	
12	\$.ajax({url:"info.php", success: function (data){	
13	\$("#i1").html(data);	
14	});	
15	});	
16	window.onload = function () {	
17	function A(){	
18	if (confirm("Go to gryonline.com?")){ location.href="https://gryonline.com"; }	
19	}	
20	setTimeout(() => { A(); }, "9000");	
21	}	
22	</script>	
23	</head>	
24	<body>	
25	<div id="i1">ABC</div>	
26	<div>File uploaded and saved correctly</div>	
27	<footer>Created by: podpuszczacz.com </footer>	
28	</body>	
29	</html>	

- 1) [1] W przeglądarce zainicjowano żądanie <http://wai.pg.edu.pl/test.php>
Plik test.php zawiera kod z listingu. Zakładamy pierwsze uruchomienie i przeglądarka nie posiada zapamiętanych żądanych informacji. Ile faz rozwiązywania DNS wystąpi od momentu zainicjowania żądania do momentu zakończenia obsługi zdarzenia Load:

- 1 a) 3
b) 4
c) 5
d) 6

każda zia - 95

- 2) [2] Przeglądarka otrzymała kod z listingu i przetwarza go celem wyświetlenia. Które z poniższych sformułowań w kontekście kolejności występowania działań są prawdziwe (pamięć podręczna jest wyłączona):

- 1 a) funkcja A() zostanie wywołana po wysłaniu żądania asynchronicznego
1 b) jeśli odpowiedź żądania asynchronicznego otrzymano po udzieleniu odpowiedzi Yes w funkcji A(), to nie zostanie ona wyświetlona
c) żądanie asynchroniczne jest wywoływane 1 raz, a funkcja A() wielokrotnie
d) odpowiedź żądania asynchronicznego zostanie zawsze wyświetlona przed wywołaniem funkcji A()
e) wywołana zostanie funkcja A(), a następnie żądanie asynchroniczne

każda zia - 1

- 3) [2] Zaprezentowany w linii 12 sposób dociągania treści strony
a) nie zadziała w przypadku niemożności dociągnięcia jQuery
b) nie zadziała w przypadku wyłączenia obsługi JavaScript
c) nie zadziała, jeśli będzie dołączona biblioteka jQuery w wersji slim
d) będzie działał zawsze, dlatego jest powszechnie wykorzystywany
e) nie zadziała w przypadku 404 dla żądania dociągnięcia jQuery UI

każde zia - 1

0,8
0,8
0,4

4) [1] Wskaż zdania prawdziwe co do zawartości znacznika <div id="i1"> dla tej strony. Pamięć podręczna jest dostępna, JS włączone.

- 0,5 ☐ a) zawartość tego znacznika zostanie podmieniona dociągniętą treścią za każdym odświeżeniem strony(F5)
- 0,5 ☐ b) żądanie asynchroniczne dociągające nową zawartość będzie pobierać dane z serwera mimo dostępnego cache;
- c) zawiera on ciąg znaków ABC i w takiej postaci zostanie wyświetlony
- d) żądanie asynchroniczne dociągające nową zawartość będzie pobierać dane z cache;
- e) zawartość tego znacznika zostanie podmieniona nową treścią co 9 sekund

każda zia -0,5

5) [2] Żądanie POST przesyłające uploadowany plik na serwer zakończyło się statusem 200 oraz przesłaniem do wyświetlenia strony widocznej na listingu. W jakim przypadku, dla tego żądania, wystąpi problem powtórnego przesłania danych z formularza po odświeżeniu strony (F5)?

- 0,66 ☐ a) przy pozytywnym zatwierdzeniu okna z komunikatem i wciśnięciu F5
- 0,66 ☐ b) przy negatywnym zatwierdzeniu okna z komunikatem i wciśnięciu F5
- 0,66 ☐ c) dla każdego F5 wykonanego w ciągu pierwszych 9 sekund
- ☐ d) jeśli jest wyłączona obsługa Java Script i wciśnięto F5 w dowolnym momencie
- ☐ e) dla wciśnięcia F5 po pozytywnym zatwierdzeniu okna i pod warunkiem, że wymagana do działania biblioteka JQuery UI nie została prawidłowo pobrana z serwera CDN

każda zia -1

6) [1] Unified ID 2.0 (UID2) to:

- 1 ☐ a) nowa nazwa dla ciasteczka stron trzecich (bez zmiany technologii funkcjonowania)
- ☐ b) nowa nazwa i technologia w miejsce wymierających podstawowych ciasteczek aplikacji
- ☐ c) nowy niezależny od plików cookie stron trzecich i nienaruszający prywatności identyfikator użytkownika wspierający wyświetlanie spersonalizowanych reklam
- d) nowy standard komunikacji, w miejsce protokołu HTTP, zapewniający jednolitą komunikację wielu różnych typów urządzeń z tym samym jednolitym kodem wystawionym po stronie serwerowej

każda zia -0,5

7) [1] CDN to:

- 1 ☐ a) rozproszony system serwerów, pośredników i zasad przekierowywania ruchu zapewniający wysoki i wydajny dostęp do popularnych zasobów
- b) pojedynczy serwer, na którym organizacja standaryzująca umieszcza zasoby, z których najczęściej korzystają twórcy witryn internetowych
- c) serwer podrzędny względem serwera www, umieszczony w tej samej podsieci i dedykowany do przetrzymywania zasobów, o które najczęściej pytają użytkownicy tej podsieci
- d) serwer podrzędny względem serwera www, umieszczony w tej samej podsieci. Jest wydzielony by przechowywać i dostarczać pliki wspólne (np jQuery.1.14.min.js) dla wszystkich witryn osadzonych na serwerze www z lokalnej podsieci

każda zia -0,5

8) [1] W protokole HTTP, do kategorii nagłówków dotyczących żądania (*request header fields*) zalicza się:

- 1 ☐ a) Authorization, If-Modified-Since, Referer
- c) Location, WWW-Authenticate
- b) Authorization, Referer, Date
- d) Location, Content-Type

każda zia -0,5

9) [1] Moduł ModeRewrite - (zaznacz prawdziwe zdania):

- 0,5 ☐ a) służy do dynamicznego przepisywania żądania URL na inną formę;
- b) jest odpowiedzialny za zarządzanie mechanizmem sesji;
- c) jest odpowiedzialny za prawidłowe zwalnianie połączenia do bazy danych
- 0,5 ☐ d) jest modułem, który rozszerza podstawowe działanie serwera WWW, takiego jak Apache

każda zia -0,5

[B_2] WAI 05.02.2025 Imię, nazwisko, indeks:
W pytaniu testowym za oznaczenie złej odpowiedzi przydzielane jest minus 50% pkt z pytania (do zera w ramach pyt.)

10)[1] Kontroler to element wzorca MVC, który dedykowany jest do:

- 1 ☒ a) definiowania i wykonywania logiki aplikacji *każda zła -0,5*
☐ b) definiowania i wykonywania logiki widoku
☐ c) wydzielania wszelkich operacji charakterystycznych dla warstwy 3 (warstwa danych)
☐ d) obsługi renderowania widoku jaki ma być wysłany do klienta

11)[1] Mechanizm negocjowania przekazywania identyfikatora sesji przy ustawionych w php.ini parametrach o postaci:

session.use_trans_sid=1
session.use_cookies=1
session.use_only_cookies=0

dopuszcza następujące działanie/a:

każda zła -0,5

- ☐ a) będzie zawsze wykorzystane przepisywanie URL, bo jest zdefiniowane w pliku konfiguracyjnym jako pierwszy dopuszczony sposób
☐ b) kolejność zapisu ustawień w pliku nie ma znaczenia. Skoro linia 2 dopuszcza ciasteczka, a linia 1 dopuszcza przepisywanie URL to będą oba sposoby wykorzystane; całkowicie losowy wybór jednej z dostępnych metod przy inicjalizacji sesji.

- 0,5* ☒ c) jeśli przeglądarka będzie miała włączoną obsługę ciasteczek, to zostaną wynegocjowane ciasteczka
0,5 ☒ d) jeśli przeglądarka będzie miała wyłączoną obsługę ciasteczek, to zostanie wynegocjowane przepisywanie URL

12)[1] Dla użytkownika, który do tej pory nie korzystał z mechanizmu sesji komenda session_start(); spowoduje po stronie serwera

- 0,5* ☒ a) utworzenie nowego obiektu stanu sesji (o określonym id)
0,5 ☒ b) ustawienie dla odpowiedzi HTTP nagłówka set-cookie z wartością identyfikatora nowo utworzonej sesji
☐ c) ustawienie dla odpowiedzi HTTP nagłówka cookie z wartością identyfikatora nowo utworzonej sesji
☐ d) poszukanie najstarszego nieużywanego obiektu stanu sesji, oczyszczenie go automatycznym unset() i ustawienie go jako bieżącego obiektu sesyjnego dla tego użytkownika *każda zła -0,5*

13)[2] Sesja (w znaczeniu zbioru kolejnych operacji jednego określonego użytkownika) zakończy się:

- 0,66* ☒ a) po czasie określonym w session.gc_maxlifetime
0,66 ☒ b) po prawidłowym wylogowaniu z aplikacji (usunięcie obiektu sesyjnego, usunięcie ciasteczka)
0,66 ☒ c) po upływie czasu życia ciasteczka w przeglądarce zdefiniowanym w session.cookie_lifetime. Zakładamy niewykonanie prawidłowego wylogowania w aplikacji *każda zła -1*
☐ d) po wykonaniu session_unset();

14)[1] Zaznacz zdania prawdziwe dla bazy MongoDB

- 0,5* ☒ a) w ramach jednej bazy występują różne kolekcje (np. użytkownicy, produkty). Dokumenty o tym samym typie grupowane są w ramach określonej kolekcji
0,5 ☒ c) dokumenty przechowywane w ramach jednej kolekcji mogą różnić się strukturą (co do pól opcjonalnych)
☐ b) dla każdej aplikacji webowej konieczne jest zainstalowanie osobnej instancji silnika bazy danych do przechowywania danych tej aplikacji
☐ d) serwer bazodanowy musi być zainstalowany na oddzielnej maszynie fizycznej niż serwer www *każda zła -0,5*

Obróć kartkę 😊

15)[1] Które z poniższych metod pozwalają na bezpieczne przechowywanie hasła w bazie MongoDB?

- 1 ☐ a) mechanizmy ochrony danych w MongoDB zapewniają wystarczający poziom bezpieczeństwa i hasło można zapisać bezpośrednio w bazie bez dodatkowych modyfikacji;
☒ b) wykorzystanie funkcji hashującej MD5;
☒ c) wykorzystanie funkcji mieszającej bcrypt;
d) zakodowanie hasła do formatu base64 przy pomocy funkcji base64_encode(\$hasło). Podczas logowania, oryginalne hasło można odtworzyć za pomocą funkcji base64_decode i porównać bezpośrednio z hasłem przesłanym przez użytkownika: base64_decode(\$hasło_z_bazy) == \$_POST["hasło"]

hasła zła -0,5

16)[1] Jak działają *variable functions* w PHP i do czego mogą być wykorzystane?

- 0,5 ☒ a) umożliwiają wywołanie funkcji, której nazwa będzie dynamicznie wyznaczona
0,5 ☒ b) są mechanizmem odwoływania się do zmiennych bez znajomości ich nazw a priori
☒ c) są wywoływane w taki sam sposób jak zwykłe funkcje, ale z użyciem zmiennej zawierającej nazwę funkcji
d) służą do definiowania funkcji anonimowych.

hasła zła -0,5

17)[2] Wysłano żądanie o postaci "http://192.186.56.10/show?id=1238741589". Logika aplikacji odpytuje bazę danych \$product = \$db->products->findOne(['_id' => new ObjectId(\$id)]); Jednak w bazie nie istnieje produkt o tym id i zwracana jest wartość null. Właściwą reakcją na tą sytuację to:

- 2 ☐ a) exit;
☒ b) http_response_code(404);
☐ c) header(Location : /index.php);
d) echo("500 - wewnętrzny błąd serwera");

hasła zła -1

18)[1] Załóżmy, iż mamy w pliku x.php zadeklarowaną klasę Osoba o polach imię i wiek. Obiekt tej klasy jest dostępny w zmiennej \$ktos= new Osoba; ustawiono mu imię na wartość John (czyli \$ktos->imię = "John";), w analogiczny sposób wiek na 54. Obiekt \$ktos po odpowiednim spreparowaniu zostaje 'w całości' przesłany w odpowiedzi na żądanie asynchroniczne do kodu JS dostępnego po stronie klienta i umieszczony w zmiennej klient; Która z poniższych linii zapewni poprawne umieszczenie całego obiektu klient w LocalStorage:

- 1 ☐ a) localStorage["user"] = klient.all();
☒ b) localStorage.setItem("user", JSON.stringify(klient));
☐ c) localStorage.Add('user', JSON.stringify(klient));
d) localStorage.setItem('user', JSON.stringify(klient.wiek));

hasła zła -0,5

19)[1] Widok częściowy (tzw. *partial view*):

- 1 ☐ a) to widok częściowy, który zostawia "puste okienko" które wypełniane jest dopiero po stronie klienta za pomocą żądań asynchronicznych
☒ b) wydzielony widok częściowy, np. menu bocznego. Jest wydzielany do osobnego pliku, by w razie potrzeby modyfikacji tego kawałka kodu HTML nie było konieczności wielokrotnego poprawiania bazowych widoków poszczególnych podstron php
☐ c) to plik, który musi być dołączony przez require i to zaraz na początku kodu php, zanim wykonamy operację logiki biznesowej
d) żadna z pozostałych odpowiedzi nie jest poprawna

hasła zła -0,5

20)[1] Problem "Headers already sent" wystąpi, gdy po dokonaniu zapisu czegoś do bufora (w ciele paczki HTTP) wywołana zostanie funkcja

- 0,33 ☒ a) session_start()
0,33 ☒ b) header('Content-Type: application/xml')
0,33 ☐ c) http_response_code (404)
d) żadna z wymienionych

hasła zła -0,5