

Karta przedmiotu

Nazwa i kod przedmiotu	Rozwój aplikacji internetowych w medycynie, PG_00068231						
Kierunek studiów	Inżynieria biomedyczna						
Data rozpoczęcia studiów	październik 2025 r.		Rok akademicki realizacji przedmiotu		2027/2028		
Poziom kształcenia	I stopnia - inżynierskie		Grupa zajęć		Grupa zajęć fakultatywnych Grupa zajęć powiązanych z prowadzonymi badaniami naukowymi w dziedzinie nauki związanej z kierunkiem - profil ogólnoakademicki		
Forma studiów	stacjonarne		Sposób realizacji		na uczelni		
Rok studiów	3		Język wykładowy		polski		
Semestr studiów	5		Liczba punktów ECTS		3.0		
Profil kształcenia	ogólnoakademicki		Forma zaliczenia		egzamin		
Jednostka prowadząca	Wydziały Politechniki Gdańskiej -> Wydział Elektroniki, Telekomunikacji i Informatyki -> Katedra Inżynierii Biomedycznej						
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot		dr inż. Anna Węsierska				
	Prowadzący zajęcia z przedmiotu						
Formy zajęć	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	30.0	0.0	0.0	15.0	0.0	45
	W tym liczba godzin zajęć na odległość: 0.0						
Aktywność studenta i liczba godzin pracy	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów		Udział w konsultacjach		Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	45		4.0		26.0	75
Cel przedmiotu	Celem przedmiotu jest zdobycie przez studenta wiedzy i umiejętności z zakresu podstawowych metod i technik tworzenia aplikacji internetowych oraz ich praktycznego zastosowania w ramach realizowanego projektu.						

Efekty uczenia się przedmiotu	Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
	[K6_W04] zna i rozumie w zaawansowanym stopniu zasady, metody i techniki programowania oraz zasady tworzenia oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, specyficznych dla kierunku studiów, a także organizację pracy systemów wykorzystujących komputery lub te urządzenia	Student zdobył wiedzę w zakresie: - instalacji i konfiguracji środowiska programistycznego dla wybranych języków programowania (np. Python Flask, Java, JavaScript, PHP, HTML); - instalacji i konfiguracji środowiska serwera WWW wraz z podstawowymi usługami; - tworzenia programów uruchamianych w środowisku przeglądarki internetowej; - tworzenia aplikacji opartych na modelu klient-serwer, w tym programowania zarówno po stronie klienta, jak i serwera; - znajomości typów architektur opartych na modelu klient-serwer, takich jak architektura bezserwerowa czy architektura mikrousług; - programowania generującego dynamiczną treść stron internetowych; - synchronizacji zasobów współdzielonych w środowiskach wielowątkowych; - synchronizacji wątków w programowaniu wielowątkowym; - integracji aplikacji internetowych z bazami danych; - zastosowania rozwiązań wykorzystujących sztuczną inteligencję w aplikacjach internetowych (np. chatboty, personalizacja treści i usług); - wykorzystania narzędzi AI w procesie tworzenia aplikacji internetowych.	[SW1] Ocena wiedzy faktograficznej
	[K6_U04] potrafi wykorzystywać posiadaną wiedzę z zakresu metod i technik programowania oraz dobrać i zastosować właściwe metody i narzędzia programistyczne w tworzeniu oprogramowania komputerów albo programowania urządzeń lub sterowników wykorzystujących mikroprocesory albo elementy lub układy programowalne, charakterystycznych dla danego kierunku studiów	Efektom kształcenia jest nabycie umiejętności: - zainstalowania i skonfigurowania środowiska programistycznego dla języka programowania (np. Python Flask, Java, JavaScript, PHP, HTML), - zainstalowania i skonfigurowania środowiska serwera WWW z podstawowymi usługami, - napisania programu w wykorzystującego model klient-serwer (np. Python Flask, Java, JavaScript, PHP, HTML), - napisania programu uruchamianego w środowisku przeglądarki WWW, - napisania prostego programu generującego dynamiczną zawartość stron serwisów WWW, - napisania prostego programu wykorzystującego przetwarzanie równoległe (np. Wielowątkowość). - napisanie prostego programu integrującego aplikację internetową z bazą danych; - napisanie prostego programu pozwalającego na zastosowanie rozwiązań wykorzystujących sztuczną inteligencję w aplikacjach internetowych;	[SU1] Ocena realizacji zadania

Wykład:

1. Przykłady zastosowań aplikacji internetowych w medycynie.
2. Podstawowe wymagania względem aplikacji internetowych w medycynie.
3. Omówienie trendów w rozwoju aplikacji internetowych, szczególnie w obszarze medycznym.
4. Technologie i środowiska internetowe (np. Node.js, Python Flask, Python Django)
5. Przykłady usługodawców chmurowych i ich współpraca z wybranymi technologiami internetowymi (np. Heroku; Google Cloud Platform)
6. Omówienie procesu udostępnienia aplikacji użytkownikom (deployment). Konfiguracja Środowiska. Uruchomienie serwera aplikacji np. Gunicorn.
7. Model klient-serwer.
8. Programowania aplikacji po stronie klienta np. JavaScript.
9. Programowania aplikacji po stronie serwera np. JavaScript.
10. Dynamiczne treści stron internetowych.
11. Przegląd architektury w modelu klient-serwer : np. Architektura bezserwerowa; architektura mikrousług;
12. Systemy zarządzania treścią (CMS) w serwisach WWW np. WordPress.
13. Technologie baz danych wykorzystywanych w serwisach WWW.
14. Programowanie równoległe w aplikacjach internetowych (wieloprosesorowość a wielowątkowość).
15. Wykorzystanie wielowątkowości w aplikacjach internetowych;
16. Synchronizacja zasobów współdzielonych w środowiskach wielowątkowych;
17. Synchronizacji wątków w programowaniu wielowątkowym;
18. Integracja aplikacji internetowych z bazami danych;
19. Wymagania względem aplikacji internetowych krytycznych. Przykłady aplikacji internetowych czasu rzeczywistego i ich zastosowanie w medycynie.
20. Oprogramowanie strumieniowego przesyłania wideo w witrynie internetowej.
21. Zastosowanie rozwiązań wykorzystujących sztuczną inteligencję w aplikacjach internetowych (np. chatboty, personalizacja treści i usług);
22. Wykorzystanie narzędzi AI w procesie tworzenia aplikacji internetowych.

Projekt:

	1. Określenie wymagań względem przykładowej aplikacji internetowych w medycynie 2. Implementacja dynamicznej strony internetowej dla przykładowej aplikacji medycznej w wybranej technologii 3. Implementacja dynamicznych treści stron internetowych 4. Integracja aplikacji internetowych z bazami danych 5. Implementacja zaawansowanych funkcji w aplikacjach internetowych (np. wykorzystanie wielowątkowości, przetwarzanie/ strumieniowanie danych w czasie rzeczywistym, wykorzystanie narzędzi sztucznej inteligencji) 6. Hosting aplikacji internetowej		
Wymagania wstępne i dodatkowe	Metody i techniki programowania 1. Budowa programu w programowaniu strukturalnym 1.1. Zmienne, typy danych, funkcje 1.2. Instrukcje sterujące 1.3. Kompilacja i wykonywanie programów 1.4. Podstawowe struktury danych 1.5. Umiejętność przejścia od pomysłu, przez algorytm do programu 2. Budowa programu w programowaniu obiektowym 2.1. Projektowanie i zapis klas 2.2. Tworzenie i wykorzystywanie obiektów 2.3. Elementy paradygmatu obiektowego (abstrakcja, hermetyzacja, dziedziczenie, polimorfizm) 2.4. Wykorzystywanie bibliotek klas 2.4 Języki programowania wysokiego poziomu		
Sposoby i kryteria oceniania osiągniętych efektów uczenia się	Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
	Projekt	51.0%	60.0%
	Wykład - kolokwium/egzamin	51.0%	40.0%
Zalecana lista lektur	Podstawowa lista lektur	Relan, Kunal. "Building REST APIs with Flask." <i>Building REST APIs with Flask</i> (2019), Rhodes, Brandon, and John Goerzen. <i>Foundations of Python network programming</i>. Apress, 2014. Reilly, David, and Michael Reilly. <i>Java network programming and distributed computing</i>. Addison-Wesley Professional, 2002 Ballard P., Moncur M., Ajax, JavaScript i PHP. Intensywny trening, Helion 2009 Eckel B., Thinking In Java, edycja polska, Helion 2006 W3C, Rekomendacje XML i HTML, www.w3.org Welling L., Thomson L., PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty, Helion 2005	
	Uzupełniająca lista lektur	Nie ma wymagań	
	Adresy eZasobów		
Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania			
Zajęcia praktyczne w ramach przedmiotu	Nie dotyczy		

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.