

# KARTA PRZEDMIOTU (SYLABUS)

## Opis przedmiotu

Kod przedmiotu		Nazwa przedmiotu	Back-end i frameworki webowe	
AIwB/O/II/ST/B1-20			Back-end and Web frameworks	
Język wykładowy		Polski		
Rok akademicki		2026/2027		
Kierunek		Sztuczna Inteligencja w Biznesie		
w zakresie		-		
Poziom studiów		studia drugiego stopnia		
Profil studiów		ogólnoakademicki		
Forma studiów		studia stacjonarne		
Semestr / semestry		semestr trzeci		
Przynależność do grupy zajęć		B. Grupa zajęć kierunkowych B1. Grupa zajęć kierunkowych obowiązkowych		
Status przedmiotu		Obowiązkowy		
Formy realizacji zajęć dydaktycznych, wymiar, punkty ECTS		Forma zajęć	Liczba godzin zajęć dydaktycznych	Liczba punktów ECTS
		Wykład	15 [h]	3 ECTS
		Ćwiczenia	[h]	
		Konwersatorium	[h]	
		Laboratorium	30 [h]	
Powiązanie przedmiotu	z profilem studiów	Związany z prowadzoną działalnością naukową w dyscyplinie Informatyka techniczna i telekomunikacja		2 ECTS
	z uprawnieniami			ECTS
	z dyscypliną	Informatyka techniczna i telekomunikacja		3 ECTS
Forma nauczania		Tradycyjna - zajęcia zorganizowane w Uczelni		
Wymagania wstępne		Znajomość podstawowej obsługi komputera niezbędna dla wykonania ćwiczeń laboratoryjnych.		
Jednostka prowadząca		Katedra Biznesu i Finansów Międzynarodowych		
Koordynator		Dr inż. Jacek Wołoszyn		
Adres strony internetowej pjo		http://weif.uniwersytetradom.pl		
Adres e-mail, telefon koordynatora		Jacek.woloszyn@urad.edu.pl (48) 361-7410		

EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE, REALIZACJA ZAJĘĆ DYDAKTYCZNYCH, WERYFIKACJA EFEKTÓW UCZENIA SIĘ

Cel kształcenia:	<p>Celem przedmiotu jest zapoznanie studentów z zasadami tworzenia aplikacji webowych po stronie serwera (back-end) z wykorzystaniem nowoczesnych frameworków. Szczególny nacisk położony jest na praktyczną naukę pracy z frameworkiem Django – od podstawowej konfiguracji projektu po tworzenie zaawansowanych funkcjonalności i integrację z bazą danych.</p>
Treści programowe:	<p>Treści zajęć są powiązane z prowadzonymi badaniami naukowymi.</p> <p><b>Treści wykładów:</b></p> <p>W ramach wykładu omawiane są:</p> <ul style="list-style-type: none"> <li>• architektura aplikacji webowych i rola warstwy back-end,</li> <li>• podstawy HTTP, routing, przetwarzanie żądań i odpowiedzi, REST API,</li> <li>• przegląd frameworków webowych (Django, Flask, FastAPI) – porównanie możliwości,</li> <li>• szczegółowa praca z Django:             <ul style="list-style-type: none"> <li>○ struktura projektu i aplikacji Django,</li> <li>○ system szablonów (Django Templates),</li> <li>○ modelei ORM (Object-Relational Mapping),</li> <li>○ tworzenie widoków, formularzy, panel administracyjny,</li> <li>○ autoryzacja, rejestracja użytkowników, sesje,</li> <li>○ tworzenie i dokumentacja REST API z Django REST Framework,</li> <li>○ testowanie i debugowanie aplikacji,</li> <li>○ podstawy wdrażania aplikacji (np. na Heroku, Railway, VPS).</li> </ul> </li> </ul> <p>Suma: 15 [h]</p> <p><b>Treść laboratoriów:</b></p> <ol style="list-style-type: none"> <li>1. Architektura aplikacji webowych i rola warstwy back-end             <ul style="list-style-type: none"> <li>○ Definicja aplikacji webowej.</li> <li>○ Architektura klient-serwer.</li> <li>○ Rozróżnienie między front-endem a back-endem.</li> <li>○ Rola warstwy back-end w aplikacjach webowych (przechowywanie danych, logika aplikacji, autoryzacja użytkowników, zarządzanie sesjami).</li> <li>○ Połączenie front-endu z back-endem (np. przez API).</li> </ul> </li> <li>2. Podstawy HTTP, routing, przetwarzanie żądań i odpowiedzi, REST API             <ul style="list-style-type: none"> <li>○ Podstawy HTTP: metoda HTTP (GET, POST, PUT, DELETE), kody odpowiedzi HTTP (200, 404, 500), nagłówki HTTP.</li> <li>○ Routing w aplikacjach webowych.</li> <li>○ Tworzenie żądań HTTP (np. w Django, FastAPI).</li> <li>○ Obsługa odpowiedzi HTTP.</li> <li>○ Tworzenie REST API w Django (Django REST Framework) – zasady, format JSON, serializatory, widoki API.</li> </ul> </li> <li>3. Frameworki webowe – przegląd (Django, Flask, FastAPI)             <ul style="list-style-type: none"> <li>○ Krótkie wprowadzenie do frameworków Django, Flask i FastAPI.</li> <li>○ Różnice w architekturze i podejściu do tworzenia aplikacji webowych.</li> <li>○ Kiedy wybrać Django, Flask lub FastAPI – porównanie wydajności, funkcji i zastosowań.</li> <li>○ Zastosowanie frameworków do tworzenia prostych aplikacji.</li> </ul> </li> <li>4. Szczegółowa praca z Django             <ul style="list-style-type: none"> <li>○ Struktura projektu i aplikacji Django – jak wygląda struktura folderów w Django, pliki konfiguracyjne, aplikacje Django.</li> </ul> </li> </ol>

	<ul style="list-style-type: none"> <li>System szablonów (Django Templates) – używanie szablonów w Django do renderowania dynamicznych stron HTML.</li> <li>Modele i ORM – definiowanie modeli w Django i korzystanie z ORM do komunikacji z bazą danych.</li> <li>Widoki i formularze – tworzenie widoków (views) oraz formularzy, obsługa przesyłanych danych z formularzy.</li> <li>Panel administracyjny – wykorzystanie panelu administracyjnego Django do zarządzania danymi aplikacji.</li> <li>Autoryzacja, rejestracja użytkowników, sesje – implementacja systemu rejestracji, logowania, zarządzania sesjami użytkowników.</li> <li>Tworzenie i dokumentacja REST API z Django REST Framework – tworzenie API w Django i jego dokumentacja.</li> <li>Testowanie i debugowanie aplikacji – techniki testowania aplikacji webowych, korzystanie z narzędzi do debugowania.</li> <li>Wdrażanie aplikacji – podstawy wdrażania aplikacji na platformach takich jak Heroku, Railway czy VPS.</li> </ul> <p>5. Podstawy wdrażania aplikacji</p> <ul style="list-style-type: none"> <li>Wdrażanie aplikacji na platformach chmurowych (Heroku, Railway).</li> <li>Przygotowanie aplikacji do produkcji: konfiguracja serwera, bazy danych.</li> <li>Użycie narzędzi do ciągłej integracji i dostarczania (CI/CD).</li> <li>Podstawowe zabezpieczenia aplikacji przed atakami (np. zabezpieczenie przed SQL Injection, XSS).</li> </ul> <p>Suma: 30 [h]</p>
Metody dydaktyczne (kształcenia):	<ul style="list-style-type: none"> <li>- metody podające (wykład informacyjny),</li> <li>- metody programowane (z wykorzystaniem komputera),</li> <li>- Obserwacja</li> </ul>
	<p>Warunkiem zaliczenia przedmiotu jest osiągnięcie wszystkich wymaganych efektów uczenia się określonych dla przedmiotu. Uzyskanie pozytywnych ocen ze wszystkich form zajęć wchodzących w skład danego przedmiotu jest równoznaczne z jego zaliczeniem i zdobyciem przez studenta liczby punktów ECTS przyporządkowanej temu przedmiotowi. Sposób obliczenia oceny końcowej z przedmiotu określony został zarządzeniem Rektora URad.</p> <p>Sposób obliczania oceny z poszczególnych form zajęć przedstawia się następująco:</p> <p>Ocena z laboratorium: test lub projekt</p> <p>Na ocenę z wykładu składa się wynik otwartego testu pisemnego.</p>

Efekty uczenia się dla przedmiotu w odniesieniu do efektów kierunkowych i formy zajęć				Metody weryfikacji efektów uczenia się	
Numer efektu uczenia się	Opis efektów uczenia się dla przedmiotu (PEU) Student, który zaliczył przedmiot (W) zna i rozumie/ (U) potrafi /(K) jest gotów do:	Kierunkowy efekt uczenia się (KEU)	Forma zajęć	Forma weryfikacji (zaliczeń)	Metody sprawdzania i oceny
W1	zna różnice między popularnymi frameworkami back-endowymi,	K_W09 K_W10	wykład	Zaliczenie na ocenę	pisemny test otwarty
W2	rozumie zasady działania aplikacji webowych po stronie serwera,	K_W09 K_W10	wykład	Zaliczenie na ocenę	pisemny test otwarty
U1	potrafi samodzielnie stworzyć kompletną aplikację webową w Django,	K_U02 K_U04	laboratorium	Zaliczenie na ocenę	pisemny test lub projekt

U2	potrafi modelować dane, projektować widoki oraz obsługiwać logikę biznesową,	K_U02 K_U04	laboratorium	Zaliczenie na ocenę	pisemny test lub projekt
U3	umie zbudować i udostępnić API oraz połączyć aplikację z bazą danych,	K_U02 K_U04	laboratorium	Zaliczenie na ocenę	pisemny test lub projekt
K1	zna dobre praktyki organizacji kodu, bezpieczeństwa i testowania aplikacji webowych	K_K02	Wykład/ laboratorium	Zaliczenie na ocenę	Obserwacja, aktywność na zajęciach obserwacja

Literatura i pomoce naukowe	
<p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. Cholewa, J. (2023). Analiza porównawcza i wydajnościowa wybranych frameworków stosowanych przy tworzeniu zaplecza aplikacji internetowych. Uniwersytet Jagielloński w Krakowie.</li> <li>2. Michałowski, M. (2020). Tworzenie aplikacji internetowych z użyciem Django i Flask. Wydawnictwo Helion.</li> <li>3. Zawisza, P. (2021). Podstawy frameworków back-endowych: Django, Flask, FastAPI. Wydawnictwo PWN.</li> <li>4. Nowak, A. (2022). Praktyczne zastosowanie frameworków webowych w tworzeniu aplikacji back-endowych. Wydawnictwo BookMaster.</li> <li>5. Kowalski, K. (2023). Wydajność frameworków webowych: Django, Spring, Laravel - analiza porównawcza. Wydawnictwo Naukowe PWN.</li> </ol> <p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. Williams, R. (2021). Mastering Django: Core Concepts and Best Practices for Building Web Applications. Packt Publishing.</li> <li>2. Mitchell, A., &amp; Smith, D. (2022). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.</li> <li>3. Johnson, J. (2023). Backend Development with Node.js: Building and Deploying Modern Web Applications. Apress.</li> <li>4. Wołoszyn, J. W., &amp; Molga, A. M. (2025). Artificial intelligence in science and technology : from biomedical image analysis to engineering and digital security. W Monografie - Uniwersytet Technologiczno-Humanistyczny im. Kazimierza Pułaskiego (No. 346; s. 113). Uniwersytet Radomski im. Kazimierza Pułaskiego. <a href="https://katalog.uniwersytetradom.pl/1783601774065/woloszyn-jacek/artificial-intelligence-in-science-and-technology?bibFilter=178">https://katalog.uniwersytetradom.pl/1783601774065/woloszyn-jacek/artificial-intelligence-in-science-and-technology?bibFilter=178</a></li> </ol> <p>Szczegółowy wykaz dodatkowych źródeł i pomocy naukowych na pierwszych zajęciach podają prowadzący.</p>	

Naład pracy studenta potrzebny do osiągnięcia zakładanych efektów uczenia się – bilans punktów ECTS		
Udział w zajęciach, aktywność	Obciążenie studenta [h]	
	Praca własna studenta - zajęcia bez nauczyciela (ZBN)	Zajęcia dydaktyczne
Udział w wykładach i laboratoriach	X	45 [h]
Przygotowanie do zajęć, Przygotowanie do zaliczenia	30 [h]	X
Sumaryczne obciążenie pracą studenta	30 [h]/ 1,2 ECTS	45 [h]/ 1,8 ECTS
Punkty ECTS za przedmiot	3 ECTS	

Informacje dodatkowe, uwagi
<p>W przypadku studentów ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych, określone powyżej (w karcie) metody i formy weryfikacji efektów uczenia się dostosowuje się odpowiednio do indywidualnych potrzeb tych studentów.</p> <p>Szczegółowe zasady i formy wsparcia studentów ze szczególnymi potrzebami: w tym z niepełnosprawnością, przewlekle chorych podczas zajęć, zaliczeń i egzaminów określono w: Regulaminie Studiów, Zasadach Studiowania, Procedurze dotyczącej zapewnienia dostępności procesu kształcenia studentom ze szczególnymi potrzebami, w tym: z niepełnosprawnością, przewlekle chorych.</p>

