

SYLABUS

DOTYCZY CYKLU KSZTAŁCENIA 2023-2025

Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>przedmiot obieralny 2: projektowanie aplikacji webowych opartych o mikroustugi</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Instytut Informatyki, Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia II stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok II, semestr 3</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy do wyboru</i>
Język wykładowy	<i>język polski</i>
Koordinator	<i>dr inż. Piotr Lasek</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Piotr Lasek</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Projekt	Liczba pkt. ECTS
3	15			30					4

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

Aplikacje internetowe, znajomość wybranej technologii do implementacji aplikacji webowych (np. Java, .Net)

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE**3.1 Cele przedmiotu**

C ₁	Zapoznanie się z podstawowymi pojęciami, cechami i zaletami architektur systemów webowych opartych o mikrousługi, takimi jak niezależne wdrażanie, elastyczność, bezpieczeństwo, monitorowanie.
C ₂	Opanowanie wiedzy i wypracowanie umiejętności pozwalających na modelowanie mikrousług przy użyciu techniki Domain Driven Design.
C ₃	Zapoznanie się z metodami przechodzenia z klasycznych aplikacji monolitycznych na architekturę monolityczną.
C ₄	Zapoznanie się z wzorcami związanymi z przepływem pracy (orkiestracja, choreografia, wzorzec saga) oraz z rolą ciągłej integracji i ciągłego dostarczania w kontekście aplikacji mikrousług.

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu Student	Odniesienie do efektów kierunkowych
EK_01	Student zna podstawowe pojęcia i problemy związane z tworzeniem oprogramowania w architekturze mikrousługowej.	K_Wo1
EK_02	Student zna i poprawnie identyfikuje technologie i narzędzia do tworzenia systemów opartych o architekturę mikrousługową.	K_Wo2
EK_03	Student potrafi poprawnie zaimplementować wskazany zakres funkcjonalności systemu przy użyciu w formie mikrousługi korzystając przy tym z różnorodnych źródeł dokumentacji.	K_Uo1, K_Uo2
EK_04	Student potrafi zaprojektować system składający się z kilku mikrousług oraz jest w stanie skomunikować wszystkie komponenty podstawowy takiego systemu (brama, rejestr mikrousług, mikrousługi, i inne)	K_Wo1, K_Uo2, K_Ko1
EK_05	Student rozumie wyzwania i potrzebę stałego śledzenia najnowszych osiągnięć z mikrousługami za pomocą których można tworzyć nowoczesne i skalowalne systemy webowe.	K_Wo3, K_Ko1

3.3 Treści programowe**A. Problematyka wykładu**

1. Kluczowe pojęcia dotyczące mikrousług (niezależne wdrażanie, rozmiar, elastyczność, dopasowanie do organizacji zespołu)
2. Rola i zalety architektury monolitycznej (monolity jednoprosesorowe, modułowe, rozproszone)
3. Cechy architektur mikrousługowych (bezpieczeństwo, monitorowanie, spójność danych, testowanie)

4. Definiowanie granic mikrousługi (spójność (cohesion), sprzężenie (coupling))
5. Projektowanie oparte o domenę (Domain Driven Design, Agregaty, modele współdzielone, Event Storming)
6. Wzorce projektowe Do not Repeat Yourself, formaty serializacji, kontrakt (zmiany kontraktu, wersjonowanie, wczesne wykrywanie zmian)
7. Migracja monolitu do mikrousług
8. Sposoby komunikacji mikrousług (komunikacja synchroniczna i asynchroniczna, kolejki i brokery wiadomości)
9. Przepływ pracy (orkiestracja i choreografia, rola transakcji bazodanowych, wzorzec Saga)
10. Rola ciągłej integracji i ciągłego dostarczania (zarządzanie kodem źródłowym, wzorzec Monorepo i Multirepo, automatyczne wdrażanie mikrousług)
11. Integrowanie mikrousług (integracja bazodanowa, choreografia i orkiestracja, prawo Postel'a, integracja z oprogramowaniem zewnętrznym, wzorzec Strangler)
12. Modele interfejsów użytkownika w kontekście systemów opartych o mikrousługi (frontend monolityczny, mikrofrontendy, wzorzec bramy agregującej, wzorzec BFF)
13. Konteneryzacja mikrousług (maszyny fizyczne, wirtualne, kontenery, PaaS, SaaS)
14. Wprowadzenie do Kubernetes i orkiestracji kontenerów

B. Problematyka ćwiczeń laboratoryjnych

1. Tworzenie podstawowej mikrousługi (definiowanie API, porty, przechowywanie danych)
2. Persystencja danych i stosowanie mechanizmów cachowania na przykładzie wybranego silnika bazy danych frameworka do obsługi pamięci podręcznej (np. Caffeide)
3. Wykrywanie i rejestracja serwisów z użyciem serwera Eureka
4. Konfiguracja i uruchomienie API Gateway
5. Komunikacja pomiędzy mikrousługami - wykorzystanie brokera wiadomości RabbitMQ.
6. Konteneryzacja mikrousług przy użyciu silnika Docker.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Ćwiczenia laboratoryjne: indywidualne i grupowe rozwiązywanie zadań z użyciem komputera

4. METODY I KRYTERIA OCENY

4.1 Sposoby weryfikacji efektów uczenia się

Symbol efektu	Metody oceny efektów uczenia się (np.: kolokwium, egzamin ustny, egzamin pisemny, projekt, sprawozdanie, obserwacja w trakcie zajęć)	Forma zajęć dydaktycznych (w, ćw, ...)
EK_01	Test wiedzy	wykład
EK_02	Test wiedzy, obserwacja w trakcie zajęć	wykład

EK_03	kolokwium, obserwacja w trakcie zajęć	laboratorium
EK_04	kolokwium, obserwacja w trakcie zajęć	laboratorium
EK_05	Test wiedzy	wykład

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Zaliczenie efektów EK_01, EK_02, EK_05

W celu uzyskania zaliczenia z wykładu, student rozwiązuje kolokwium testowe na poziomie minimum 50%. Rozwiązując je student wykazuje, że zna pojęcia i problemy związane z tworzeniem oprogramowania w architekturze mikrousługowej i identyfikuje technologie i narzędzia do tworzenia systemów opartych o tę architekturę. Wykazuje również, że rozumie wyzwania i potrzebę stałego śledzenia najnowszych osiągnięć w zakresie stosowania mikrousług, za pomocą których można tworzyć nowoczesne i skalowalne systemy webowe. Student może uzyskać dodatkowe punkty (procenty) na podstawie zadań domowych, które będą zadawane na zajęciach wykładowych. Punkty te zostaną doliczone do wyniku testu. Ocena z wykładu: zal/nzal.

Zaliczenie efektów EK_03, EK_04

Student realizując kolejne laboratoria buduje mały system oparty o architekturę mikrousługową. Na ostatnich zajęciach tworzy jego dokumentację, która wraz z kodem źródłowym stanowi podstawę zaliczenia weryfikując tym samym, że student potrafi poprawnie zaimplementować wskazany zakres funkcjonalności systemu w architekturze mikrousługowej.

Do oceny, prowadzący zajęcia bierze następujące elementy:

- kompletność, przejrzystość i jakość kodu źródłowego (jedna osoba powinna zaimplementować przynajmniej 2 różne mikrousługi oraz wszystkie inne wymagane komponenty systemu (max. 60 punktów)
- jakość dokumentacji: kompletność, jasność, czytelność (40 punktów)

Ocena ostateczna ustalana jest na podstawie poniższej tabeli.

Uzyskany wynik [%]	Ocena
0 - 49	niedostateczna
50 - 59	dostateczna
60 - 69	dostateczna plus
70 - 79	dobra
80 - 89	dobra plus
90 - 100	bardzo dobra

5. CAŁKOWITY NAKŁAD PRACY STUDENTA POTRZEBNY DO OSIĄGNIĘCIA ZAŁOŻONYCH EFEKTÓW W GODZINACH ORAZ PUNKTACH ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny kontaktowe wynikające z harmonogramu studiów	45
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	3
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	55
SUMA GODZIN	103
SUMARYCZNA LICZBA PUNKTÓW ECTS	4

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

LITERATURA PODSTAWOWA:
1. Newman Sam, Budowanie mikrouslug. Projektowanie drobnoziarnistych systemów, Helion 2022
LITERATURA UZUPEŁNIAJĄCA:
1. Ronnie Mitra, Irakli Nadareishvili, Mikrouslugi. Budowa i działanie, APN Promise, 2021
2. Richardson Chris, Mikroserwisy, Wydawnictwo Naukowe PWN, 2020