

SYLABUS**DOTYCZY CYKLU KSZTAŁCENIA 2021-2025**

Rok akademicki 2024/2025

1. PODSTAWOWE INFORMACJE O PRZEDMIOCIE

Nazwa przedmiotu	<i>przedmiot obieralny 3: aplikacje internetowe 2</i>
Kod przedmiotu	
Nazwa jednostki prowadzącej kierunek	<i>Kolegium Nauk Przyrodniczych</i>
Nazwa jednostki realizującej przedmiot	<i>Kolegium Nauk Przyrodniczych</i>
Kierunek studiów	<i>informatyka</i>
Poziom studiów	<i>studia I stopnia</i>
Profil	<i>ogólnoakademicki</i>
Forma studiów	<i>stacjonarne</i>
Rok i semestr/y studiów	<i>rok IV, semestr 7</i>
Rodzaj przedmiotu	<i>przedmiot kierunkowy inżynierski</i>
Język wykładowy	<i>polski</i>
Koordynator	<i>dr inż. Piotr Lasek</i>
Imię i nazwisko osoby prowadzącej / osób prowadzących	<i>dr inż. Piotr Lasek, mgr inż. Jaromir Sarzyński</i>

1.1. Formy zajęć dydaktycznych, wymiar godzin i punktów ECTS

Semestr (nr)	Wykł.	Ćw.	Konw.	Lab.	Sem.	ZP	Prakt.	Inne (jakie?)	Liczba pkt. ECTS
7	10			20					3

1.2. Sposób realizacji zajęć

zajęcia w formie tradycyjnej

1.3 Forma zaliczenia przedmiotu (z toku)

zaliczenie z oceną

2. WYMAGANIA WSTĘPNE

programowanie obiektowe, inżynieria oprogramowania, bazy danych, aplikacje internetowe 1.

3. CELE, EFEKTY UCZENIA SIĘ, TREŚCI PROGRAMOWE I STOSOWANE METODY DYDAKTYCZNE

3.1 Cele przedmiotu

C ₁	Zapoznanie studentów z podstawowymi zagadnieniami związanymi z protokołami komunikacji i standardami z perspektywy aplikacji internetowych (HTTP, http 2, Rest, SSH, HTTPS, itp.)
C ₂	Nabycie umiejętności w projektowaniu i implementacji komunikacji w aplikacjach webowych (frontend – backend, Rest-full, itp.)
C ₃	Zapoznanie z narzędziami (wykorzystanie frameworka) do budowy backend'u aplikacji internetowych oraz zastosowanie ich w praktyce
C ₄	Zapoznanie z narzędziami (wykorzystanie frameworka/bibliotek) do budowy frontend'u aplikacji internetowych oraz zastosowanie ich w praktyce

3.2 Efekty uczenia się dla przedmiotu

EK (efekt uczenia się)	Treść efektu uczenia się zdefiniowanego dla przedmiotu	Odniesienie do efektów kierunkowych
EK_01	Zna podstawowe metody i techniki związane z tworzeniem aplikacji internetowych, potrafi omówić podstawowe zagadnienia i wskazać perspektywy ich rozwoju.	K_Wo4, K_Wo7
EK_02	Umie stosować podstawowe techniki i dobierać optymalne rozwiązania do budowy aplikacji internetowych przy dostępnych na rynku narzędzi.	K_U20
EK_03	Potrafi sporządzić czytelną dokumentację aplikacji webowej.	K_U21

3.3 Treści programowe

A. Problematyka wykładu

Omówienie protokołu HTTP.
Omówienie podstawowych zagadnień stylu REST.
Podstawowe komponenty budowy aplikacji webowych (model MVC).
Wykorzystanie SpringBoot do implementacji aplikacji webowych.
Projektowanie kontrolerów RESTful.

B. Problematyka ćwiczeń laboratoryjnych

Przegląd technologii (bibliotek/framework'ów) wykorzystywanych do budowy aplikacji internetowych.
Tworzenie aplikacji internetowych zgodnie z stylem REST API oraz dobrymi praktykami sugerowanymi przy ich tworzeniu. Tworzenie dokumentacji API.
Wybrane zagadnienia dotyczące aplikacji internetowych odnośnie backend'u aplikacji.
Wybrane zagadnienia dotyczące aplikacji internetowych odnośnie frontend'u aplikacji.
Realizacja projektów zaliczeniowych.

3.4 Metody dydaktyczne

Wykład: wykład z prezentacją multimedialną

Ćwiczenia laboratoryjne: praca w grupach i indywidualna: dyskusja, realizacja zadań wg specyfikacji

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	kolokwium	wykład
EK_02	projekt	laboratorium
EK_03	dokumentacja projektu	laboratorium

4.2 Warunki zaliczenia przedmiotu (kryteria oceniania)

Wykład

Student uzyskuje zaliczenie z wykładu na podstawie wyniku kolokwium, które przeprowadzane jest w formie testu.

Efekt EK_01 jest uznany za zaliczony, gdy student uzyska przynajmniej 50% punktów. Jeżeli ten próg zostanie przekroczony, student otrzymuje ocenę 'zal.' (zaliczenie). W przypadku nieosiągnięcia wymaganego progu, student nie uzyskuje zaliczenia.

Laboratorium

Zaliczenie z laboratorium odbywa się na podstawie zrealizowanego projektu, w którym studenci rozwiązują konkretny, zadany problem. Do wykonania będzie przygotowanie części backend'owej aplikacji, udostępniającej (udokumentowane) API pozwalające na wymianę danych z wykorzystaniem formatu JSON, oraz przygotowanie części frontend'owej aplikacji (z wykorzystaniem framework'a JS) pozwalającej na korzystanie z funkcjonalności udostępnionych w części backend'owej. Ponadto na ocenę:

- 3.0 wymagane jest zrealizowanie we frontend'owej części aplikacji operacji sortowania i filtrowania,
- 4.0 wymagane jest zrealizowanie we frontend'owej części aplikacji operacji wyszukiwania oraz mechanizmu paginacji,
- 5.0 wymagane jest poszerzenie backend'owej części aplikacji o realizowanie wyżej wymienionych składowych.

Efekt EK_02 jest uznany za zaliczony, gdy projekt wykonany przez studenta spełni wymagania określone przynajmniej na ocenę „dostateczny”.

Efekt EK_03 oceniany na 'zal.'/'nzal.' jest uznany za zaliczony, gdy do projektu utworzona jest kompletna dokumentacja przedstawiająca powstałą aplikację oraz aspekty jej tworzenia.

Ocena końcowa z laboratorium jest wystawiana na podstawie oceny za efekt EK_02, pod warunkiem, że efekt EK_03 został oceniony na 'zal'.

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

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny z harmonogramu studiów	30
Inne z udziałem nauczyciela akademickiego (udział w konsultacjach, egzaminie)	-
Godziny niekontaktowe – praca własna studenta (przygotowanie do zajęć, egzaminu, napisanie referatu itp.)	45
SUMA GODZIN	75
SUMARYCZNA LICZBA PUNKTÓW ECTS	3

6. PRAKTYKI ZAWODOWE W RAMACH PRZEDMIOTU

wymiar godzinowy	-
zasady i formy odbywania praktyk	-

7. LITERATURA

Literatura podstawowa: 1) Geofrey Warin: <i>Spring MVC 4. Projektowanie zaawansowanych aplikacji WWW</i>, Helion. 2) Michał Bentkowski, Gynvael Coldwind, Artur Czyż, Rafał Janicki, Jarosław Kamiński, Adrian Michalczyk, Mateusz Niezabitowski, Marcin Piosek, Michał Sajdak, Grzegorz Trawiński, Bohdan Widła: <i>Bezpieczeństwo aplikacji webowych</i>, SECURITUM, 2019. 3) Tomasz Sochacki: <i>JavaScript. Interaktywne aplikacje webowe</i>, Gliwice, Helion, 2020. 4) Tomasz Sochacki: <i>JavaScript. Tworzenie nowoczesnych aplikacji webowych</i>, Gliwice, Helion, 2020.
Literatura uzupełniająca: • https://developer.mozilla.org/en-US/docs/Web • https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design • https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-implementation • https://restfulapi.net/

Akceptacja Kierownika Jednostki lub osoby upoważnionej