

KARTA PRZEDMIOTU**1. Dane podstawowe**

Nazwa przedmiotu	Zaawansowane aplikacje sieciowe
Nazwa przedmiotu w języku angielskim	Advanced network applications
Kierunek studiów	Informatyka
Poziom studiów (I, II, jednolite magisterskie)	II
Forma studiów (stacjonarne, niestacjonarne)	Stacjonarne
Dyscyplina	Informatyka
Język wykładowy	polski

Koordynator przedmiotu	mgr Piotr Kociuba
------------------------	-------------------

Forma zajęć (<i>katalog zamknięty ze słownika</i>)	Liczba godzin	semestr	Punkty ECTS
wykład			6
konwersatorium	30	I lub III	
ćwiczenia			
laboratorium	30	I lub III	
warsztaty			
seminarium			
proseminarium			
lektorat			
praktyki			
zajęcia terenowe			
pracownia dyplomowa			
translatorium			
wizyta studyjna			

Wymagania wstępne	<p>W1 - Podstawowa znajomość zasad funkcjonowania sieci WWW, języka HTML oraz arkuszy stylu Javascript</p> <p>W2 - Znajomość języka angielskiego w stopniu umożliwiającym czytanie dokumentacji technicznych</p> <p>W3 – znajomość obsługi baz danych</p>
-------------------	---

2. Cele kształcenia dla przedmiotu

<p>C1: Zapoznanie z podstawami języka Typescript</p> <p>C2: Zapoznanie z tworzeniem aplikacji internetowych</p> <p>C3: Zapoznanie z tworzeniem komponentów webowych</p>

3. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

Symbol	Opis efektu przedmiotowego	Odniesienie do efektu kierunkowego
WIEDZA		
W_01	Zna możliwości Typescript w kontekście wykorzystania ich jako zaplecza do tworzenia serwisu webowego	K_W01, K_W04
W_02	Zna składowe poprawnie zaprojektowanego serwisu webowego	K_W02
UMIEJĘTNOŚCI		
U_01	Zna aplikacje pozwalające na tworzenie elementów składowych witryn internetowych	K_U09
U_02	Potrafi korzystać z dokumentacji komponentów i bibliotek wykorzystywanych w procesie budowy witryny	K_U02, K_U09
U_03	Potrafi zaprojektować stronę internetową lub fronton	K_U05, K_U17
KOMPETENCJE SPOŁECZNE		
K_01	Potrafi ocenić jakość opracowanego rozwiązania i porównać z rozwiązaniami alternatywnymi	K_K01, K_K05
K_02	Potrafi wybrać tematykę i sposób realizacji projektu	K_K03, K_K06

4. Opis przedmiotu/ treści programowe

<ol style="list-style-type: none"> 1. Praca w Visual Studio Code 2. Typescript 3. Lit-Element, Lit-HTML, Vaadin komponent 4. Tworzenie własnych komponentów 6. Architektura rozwiązań fullstack Typescript 7. Aplikacje Node.js 8. Restfull API
--

5. Metody realizacji i weryfikacji efektów uczenia się

Symbol efektu	Metody dydaktyczne <i>(lista wyboru)</i>	Metody weryfikacji <i>(lista wyboru)</i>	Sposoby dokumentacji <i>(lista wyboru)</i>
WIEDZA			
W_01	Wykład konwersatoryjny	Test	Uzupełniony i oceniony test
W_02	Wykład konwersatoryjny	Test	Uzupełniony i oceniony test
UMIEJĘTNOŚCI			
U_01	Ćwiczenia laboratoryjne, metoda projektu design thinking	Przygotowanie projektu	Pliki projektu
U_02	Ćwiczenia laboratoryjne, metoda projektu design thinking	Przygotowanie projektu	Pliki projektu
U_03.	Ćwiczenia laboratoryjne, metoda projektu	Przygotowanie projektu	Pliki projektu

	design thinking		
KOMPETENCJE SPOŁECZNE			
K_01	Ćwiczenia laboratoryjne design thinking	Przygotowanie projektu	Pliki projektu
K_02	Ćwiczenia laboratoryjne design thinking	Przygotowanie projektu	Pliki projektu

6. Kryteria oceny, wagi

Konwersatorium:

Na końcową ocenę z konwersatorium składają się:

- test 90%,
- aktywny udział w dyskusji na wykładzie 10%,

Kryteria oceniania prac na teście:

- 91 - 100% punktów z testu - ocena 5,0
- 81 - 90% punktów z testu - ocena 4,5
- 71 - 80% punktów z testu - ocena 4,0
- 61 - 70% punktów z testu - ocena 3,5
- 50 - 60% punktów z testu - ocena 3,0

Laboratorium:

Na końcową ocenę z laboratorium składają się:

- oddane dwa projekty 90%,
- aktywny udział w części praktycznej laboratorium 10%,

Kryteria oceniania prac z projektu:

- 91 - 100% punktów z projektów - ocena 5,0
- 81 - 90% punktów z projektów - ocena 4,5
- 71 - 80% punktów z projektów - ocena 4,0
- 61 - 70% punktów z projektów - ocena 3,5
- 50 - 60% punktów z projektów - ocena 3,0

7. Obciążenie pracą studenta

Forma aktywności studenta	Liczba godzin
Liczba godzin kontaktowych z nauczycielem	90
Liczba godzin indywidualnej pracy studenta	70

8. Literatura

Literatura podstawowa
1. Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js w akcji, Helion 2. Nicolas Bevacqua, Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion 3. Boris Chemy, Programowanie w języku TypeScript: Tworzenie skalowalnych aplikacji w JavaScript, Wydawnictwo Naukowe PWN; 2020
Literatura uzupełniająca
1. Brad Dayley Piotr Pilch, Node.js, MongoDB, AngularJS: kompendium wiedzy, Grupa Wydawnicza Helion, Gliwice 2015