

# SYLABUS ZAJ /GRUPY ZAJ

## Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:	Inżynieria systemów inteligentnych				
Nazwa zaj / grupy zaj :	Technologie webowe w aplikacjach internetu rzeczy II				
Course / group of courses:	Web Technologies in Internet of Thing Applications II				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z-IS				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105887	Kod Erasmus:			
Punkty ECTS:	1	Rodzaj zaj :		obowiązkowy	
Rok studiów:	3	Semestr:		6	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
3	6	P	15	Zaliczenie z ocen	1
Razem			15		1
Koordinator:	magister inżynier Dariusz Piwko				
Prowadzący zajęcia:					
Język wykładowy:	semestr: 6 - język polski				

## Objaśnienia:

Rodzaj zaj : obowiązkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zajęcia seminaryjne, P - wiczenia praktyczne (w tym zajęcia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zajęcia z technologii informacyjnych, P – wiczenia projektowe, ZT – zajęcia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

## Dane merytoryczne

Wymagania wstępne:			
Wiadomości z zakresu przedmiotu "Technologie webowe w aplikacjach internetu rzeczy"			
Szczegółowe efekty uczenia się			
Lp.	Student, który zaliczył zajęcia zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia się
1	Student posiada wiedzę i umiejętności do stworzenia aplikacji internetowej opartej o poznane wzorce	IN1_U01	praca pisemna
2	Student posiada wiedzę i umiejętności do korzystania z dokumentacji technicznej bibliotek oraz pakietów. Jest w stanie stworzyć dokumentację techniczną swojego projektu.	IN1_U01, IN1_U11, IN1_U07	praca pisemna
3	Student posiada wiedzę i umiejętności do użycia poznanych języków programowania oraz bibliotek	IN1_U05	praca pisemna
4	Student ma wiadomości, jak współpracować w projekcie zespół. Potrafi planować i koordynować pracę zespół.	IN1_U13, IN1_U12	praca pisemna
Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)			

<p>metody praktyczne (Forma zajęć : Zajęcia mają charakter projektu. Głównym celem jest stworzenie aplikacji internetowej przy użyciu określonych narzędzi. Metody dydaktyczne: Postęp wniosku jest weryfikowany podczas konsultacji. Projekt jest podzielony na kilka etapów w celu lepszej motywacji studentów.)</p>	
<b>Kryteria oceny i weryfikacji efektów uczenia się</b>	
<p><b>umiejętności:</b></p> <p>ocena pracy pisemnej (ocena zadania projektowego, ocena dokumentacji technicznej)</p>	
<b>Warunki zaliczenia</b>	
<p>Projekt oceniany jest w poniższej skali (zaokrąglenie na korzyść studenta)</p> <p>Skala ocen:</p> <p>5.0 (bdb) 100% - 91%</p> <p>4.5 (+db) 90% - 81%</p> <p>4.0 (db) 80% - 71%</p> <p>3.5 (+dst) 70% - 61%</p> <p>3.0 (dst) 60% - 51%</p> <p>2.0 (ndst) &lt;50%</p>	
<b>Treści programowe (opis skrócony)</b>	
<ol style="list-style-type: none"> <li>1. Omówienie cyklu życia projektu.</li> <li>2. Wybranie i omówienie zakresu projektu.</li> <li>3. Określenie wymagań funkcjonalnych i niefunkcjonalnych.</li> <li>4. Zdefiniowanie architektury systemu.</li> <li>5. Opracowanie podstawowych diagramów opisujących system.</li> <li>6. Implementacja systemu w oparciu o przygotowanie modułów.</li> <li>7. Implementacja testów systemu.</li> <li>8. Opracowanie dokumentacji technicznej.</li> <li>9. Poznanie słownictwa technicznego przy opracowywaniu projektu.</li> </ol>	
<b>Content of the study programme (short version)</b>	
<ol style="list-style-type: none"> <li>1. Discuss project life cycle.</li> <li>2. Choose and discuss project scope.</li> <li>3. Specify functional and non-functional requirements.</li> <li>4. Define system architecture.</li> <li>5. Develop basic diagrams.</li> <li>6. Create system.</li> <li>7. Develop tests for the system.</li> <li>8. Create technical documentation.</li> <li>9. Improve technical vocabulary.</li> </ol>	
<b>Treści programowe</b>	
	Liczba godzin
Semestr: 6	
Forma zajęć : <b>wiczenia projektowe</b>	
<p>Zajęcia projektowe składają się z ogólnych zasad tworzenia oprogramowania w IT:</p> <ol style="list-style-type: none"> <li>1. Omówienie cyklu życia projektu. Prezentacja wszystkich faz i etapów projektowania IT. Omów planowania projektu, w tym ustalenie i udokumentowanie listy konkretnych celów projektu.</li> <li>2. Wybranie i omówienie zakresu projektu, dokumentując listę konkretnych zadań projektu. Użycie frameworków / bibliotek poznanych podczas zajęć.</li> <li>3. Określenie wymagań funkcjonalnych i niefunkcjonalnych systemu. Przygotowanie listy funkcji, które rozwiążą określone problemy dla użytkowników.</li> <li>4. Zdefiniowanie architektury systemu wysokiego poziomu z podziałem na aplikacje klienckie i serwerowe z bazami danych.</li> <li>5. Przygotowanie diagramów architektury pokazujących integrację między frontendem i backendem z wewnętrzną strukturą techniczną. Diagramy UML, które opisują przypadki użycia, diagram ERD opisujący bazy danych oraz opis Interfejsu aplikacji (API).</li> <li>6. Wdrożenie systemu przy użyciu określonych frameworków lub bibliotek (np. Angular). System składa się z części klienta i aplikacji serwerowej, które współpracują z bazami danych (np. PostgreSQL, MySQL, MongoDB itp.).</li> <li>7. Określenie przepływu aplikacji testującej. Głównymi celami są zintegrowane testy, które eliminują wiele wspólnych błędów.</li> </ol>	15

8. Przygotowanie dokumentacji technicznej opisującej każdą fazę tworzonej aplikacji. 9. Poznanie słownictwa technicznego przy opracowywaniu projektu.	15
<b>Literatura</b>	
Podstawowa	
Craig Walls, Spring in Action, Fifth Edition, Manning Publications 2018	
Gaston C. Hillar, Hands-On MQTT Programming with Python: Work with the lightweight IoT protocol in Python, Packt Publishing 2018	
Michał Sajdak (red), Bezpieczeństwo aplikacji webowych, Securitum Szkolenia, Kraków 2019	
Mike Cantelon, Marc Harter, TJ Holowaychuk, Nathan Rajlich, Node.js in Action, Manning Publications 2014	
Dokumentacja techniczna dostępna w sieci Internet opisująca użyte frameworki podczas zajęć	
Uzupełniająca	

#### Dane jakościowe

Przyporządkowanie zajęć /grup zajęć do dyscypliny naukowej/artystycznej		informatyka techniczna i telekomunikacja	
Sposób określenia liczby punktów ECTS			
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)		Obciążenia studenta [w godz.]	
Udział w zajęciach		15	
Konsultacje z prowadzącym		0	
Udział w egzaminie		0	
Bezpośredni kontakt z nauczycielem - inne		0	
Przygotowanie do laboratorium, ćwiczeń, zajęć		0	
Przygotowanie do kolokwium i egzaminu		0	
Indywidualna praca własna studenta z literatury, wykładami itp.		0	
Inne		10	
Sumaryczne obciążenie prac studenta		25	
Liczba punktów ECTS			
Liczba punktów ECTS		1	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego		L. godzin	ECTS
		15	0,6
Zajęcia o charakterze praktycznym		L. godzin	ECTS
		25	1,0

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zajęć wymagających bezpodrzedniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć/grup zajęć.