

Karta opisu zajęć - Sylabus

Państwowa Wyższa Szkoła Techniczno-Ekonomiczna im. ks. Bronisława Markiewicza w Jarosławiu

I. INFORMACJE PODSTAWOWE

Nazwa zajęć: Programowanie aplikacji internetowych z Node.js

Cykl kształcenia rozpoczynający się w roku akademickim 2021/2022

Nazwa kierunku studiów, poziom i profil kształcenia:
Informatyka, studia pierwszego stopnia, profil praktyczny

Język wykładowy: polski

Rodzaj zajęć: zajęcia specjalistyczne

Rok studiów: 3

Semestr: 6

Liczba punktów ECTS przypisana zajęciom: 2

Koordynator zajęć
Imię, nazwisko, tytuł/stopień naukowy, adres e-mail:
Janusz Bytnar, dr inż., janusz.bytnar@pwste.edu.pl

Jednostka organizacyjna:

FORMA PROWADZENIA ZAJĘĆ I LICZBA GODZIN

Ogólna liczba godzin zajęć dydaktycznych na studiach stacjonarnych i niestacjonarnych z podziałem na formy:

Studia stacjonarne		Studia niestacjonarne	
Wykład:		Wykład:	
Ćwiczenia:		Ćwiczenia:	
Laboratorium:		Laboratorium:	
Lektorat:		Lektorat:	
Projekt:		Projekt:	
Zajęcia praktyczne:	30	Zajęcia praktyczne:	
Seminarium:		Seminarium:	
Zajęcia terenowe:		Zajęcia terenowe:	
Praktyki zawodowe:		Praktyki zawodowe:	
Inna forma (jaka):		Inna forma (jaka):	
RAZEM:	30	RAZEM:	

II. INFORMACJE SZCZEGÓŁOWE

Wymagania wstępne i dodatkowe:

w zakresie Wiedzy i Umiejętności: Student powinien posiadać wiedzę i umiejętności w zakresie realizowanym w ramach przedmiotu „Programowanie obiektowe”, sem.3, oraz przedmiotu „Współczesne języki programowania”, sem4.

wymagania w zakresie Kompetencji społecznych:

Zrozumienie potrzeby ciągłego doksztalcania się, umiejętność pozyskiwania informacji, umiejętność samokształcania się.

Cel (cele) kształcenia dla zajęć:

Zapoznanie studentów z zakresu wykorzystania możliwości wieloplatformowego środowiska uruchomieniowego do tworzenia aplikacji typu server-side – Node.js

EFEKTY UCZENIA SIĘ OKREŚLONE DLA ZAJĘĆ I ICH ODNIESIENIE DO EFEKTÓW UCZENIA SIĘ OKREŚLONYCH DLA KIERUNKU STUDIÓW

<p>Efekty uczenia się określone dla zajęć w kategorii wiedza, umiejętności oraz kompetencje społeczne oraz metody weryfikacji efektów uczenia się</p> <p>UWAGA:</p> <p>Dzielimy efekty uczenia się określone dla zajęć na kategorie wiedzy, umiejętności i kompetencji społecznych. Określone dla zajęć efekty uczenia się nie muszą obejmować wszystkich trzech kategorii i zależą one od formy zajęć.</p>				
Symbol efektów uczenia się określonego dla zajęć*	Treść efektu uczenia się. Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się, student w kategorii:			Odniesienie do efektów uczenia się określonych dla kierunku studiów (symbol efektów uczenia się)
Wiedzy - zna i rozumie				
E_01	Możliwości programistyczne technologii Node.js			K_W08, K_W10, K_W20
E_02	Praktyczne zastosowania aplikacji wielowarstwowych wykorzystujących technologii Node.js			K_W08, K_W10, K_W20
Umiejętności - potrafi				
E_03	Zaprojektować i wykonać projekt aplikacji internetowej z wykorzystaniem Node.js			K_U02, K_U09, K_U12, K_U14, K_U20
E_04	Wykorzystać w aplikacji dowolny format bazodanowy oraz dowolny sposób przesyłania danych pomiędzy aplikacją a serwerem			K_U09, K_U12, K_U18
E_05	Zastosować w aplikacji framework Express.js lub podobny			K_U02, K_U09, K_U12, K_U14, K_U18, K_U20
Kompetencji społecznych - jest gotów do				
E_06	Podnoszenia swoich kwalifikacji zawodowych poprzez uczestnictwo w szkoleniach specjalistycznych oraz zdobywanie certyfikatów developerskich.			K_K01, K_K05
<p>UWAGA!</p> <p>Zaleca się, aby w zależności od liczby godzin zajęć, liczba efektów uczenia się zawierała się w przedziale: 3-7, ale są to wartości umowne.</p>				
TREŚCI PROGRAMOWE I ICH ODNIESIENIE DO FORM ZAJĘĆ I METOD OCENIANIA				
Treści programowe (uszczegółowione, zaprezentowane z podziałem na poszczególne formy zajęć, tj. wykład, ćwiczenia, laboratoria, projekty, seminaria i inne):				
Symbol treści programowych	Opis treści programowych	Forma zajęć	Metody dydaktyczne prowadzenia zajęć umożliwiające osiągnięcie założonych efektów uczenia się *	Metody weryfikacji osiągnięcia efektów uczenia się przypisanych do zajęć #
		zajęcia praktyczne		
TP-01	Zajęcia praktyczne dotyczące przygotowania i konfiguracji środowiska Node.js. Wstępne przygotowanie projektów aplikacji.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

TP-02	Rozbudowa aplikacji z poprzednich zajęć poprzez wykorzystanie menagera pakietów NPM. Konfiguracja projektu aplikacji dla bazy danych MySQL lub innego formatu bazodanowego. Testowanie połączenia z bazą danych		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP-03	Rozbudowa aplikacji poprzez dodanie podstawowych operacji bazodanowych (CRUD) z poziomu kodu Node.js		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP-04	Dalszy ciąg modyfikacji aplikacji poprzez wykorzystanie bardziej zaawansowanych operacji bazodanowych		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP-05	Wykorzystanie frameworka Express.js – instalacja i testowanie podstawowych formularzy		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP-06	Rozbudowa aplikacji poprzez wykorzystanie formularzy Express.js do podstawowych operacji bazodanowych CRUD		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

TP-07	Wykorzystanie technologii AJAX w aplikacjach Node.js Express. Rozbudowa wcześniej tworzonych aplikacji.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów
TP-08	Rozbudowa i udoskonalanie projektów zaliczeniowych. Obrona wykonanych w ramach zajęć projektów.		zajęcia praktyczne – realizacja projektów w postaci aplikacji	wykonanie ćwiczeń, projekt zaliczeniowy - kontrola postępów w realizacji projektów

Metody weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć, powinny być zróżnicowane w zależności od kategorii, tj. inne dla kategorii wiedza i inne dla kategorii umiejętności i kompetencje społeczne.

Dla wykładu:

* np. wykład podający, wykład problemowy, ćwiczenia oparte na wykorzystaniu różnych źródeł wiedzy

np. egzamin ustny, test, prezentacja, projekt

Zaleca się podanie przykładowych zadań (pytań) służących weryfikacji osiągnięcia efektów uczenia się określonych dla zajęć.

ZALECANA LITERATURA (w tym pozycje w języku obcym)

Literatura podstawowa (powinna być dostępna dla studenta w uczelnianej bibliotece):

1. Negrino T, Smith D.: „Po prostu JavaScript i Ajax”, Helion 2010,
2. Duckett J.: „JavaScript i jQuery. Interaktywne strony WWW dla każdego. Podręcznik Front-End Developera”, wyd. Helion, 2015,
3. Kalbarczyk D i A: „AngularJS: pierwsze kroki”, Helion 2015

Literatura uzupełniająca:

1. Brown E.: „Tworzenie aplikacji internetowych z użyciem Node i Express”, Helion 2020
2. Herron D.: „Platforma Node.js. Przewodnik webdevelopera”, Helion 2016

III. INFORMACJE DODATKOWE

BILANS PUNKTÓW ECTS

OBCIĄŻENIE PRACĄ STUDENTA (godziny)

Forma aktywności	Liczba godzin *
Godziny zajęć (według harmonogramu) z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	30
Praca własna studenta	30(czytanie literatury, praca nad realizacją projektu)
SUMA GODZIN:	60

OBCIĄŻENIE PRACĄ STUDENTA (punkty ECTS)

		Liczba punktów ECTS	
SUMARYCZNA LICZBA PUNKTÓW ECTS PRZYPIŚANYCH DO ZAJĘĆ	Praca studenta wymagająca bezpośredniego kontaktu z nauczycielem akademickim lub inną osobą prowadzącą zajęcia	Ogółem: 2	1

	Praca własna studenta		1
* godziny lekcyjne, czyli 1 godz. oznacza 45 min;			
OPIS PRACY WŁASNEJ STUDENTA:			
Praca własna studenta musi być precyzyjnie opisana, uwzględniając charakter praktyczny zajęć. Należy podać symboleffektu uczenia się, którego praca własna dotyczy oraz metody weryfikacji efektów uczenia się stosowane w ramach pracy własnej. Przykładowe formy aktywności: (1) przygotowanie do zajęć, (2) opracowanie wyników, (3) czytanie wskazanej literatury, (4) napisanie raportu z zajęć, (5) przygotowanie do egzaminu, opracowanie projektu.			
Samodzielna praca związana z czytaniem wskazanej literatury oraz przygotowanie do zaliczenia projektu (E_01 – E_02). Samodzielna realizacja ćwiczeń programistycznych na podstawie instrukcji, przygotowanie projektów indywidualnych oraz grupowych, których tematyka uzgodniona jest z prowadzącym zajęcia (E_03 – E_05).			
KRYTERIA OCENIANIA			
Na ocenę dostateczną student zrealizował projekty programistyczne dotyczące opisywanych treści programowych przedmiotu w zakresie podstawowym i potrafi omówić zastosowane techniki. Na ocenę dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór. Na ocenę bardzo dobrą student zrealizował kompletne projekty programistyczne dotyczące opisywanych treści programowych przedmiotu z zastosowaniem ponadstandardowych rozwiązań i potrafi omówić zastosowane techniki oraz uzasadnić ich wybór.			
INFORMACJA O PRZEWIDYWANEJ MOŻLIWOŚCI WYKORZYSTANIA KSZTAŁCENIA NA ODLEGŁOŚĆ			

.....
(data, podpis Koordynatora
odpowiedzialnego za zajęcia)

.....
(data, podpis Kierownika Zakładu/
Kierownika Jednostki Międzyinstytutowej)

Uwaga:
Karta opisu zajęć (sylabus) musi być dostępna dla studenta.