

Politechnika Białostocka
Wydział Informatyki

Program studiów podyplomowych

JavaScript Developer

Edycja 2021/22

Sylwetka absolwenta

Studia poświęcone są zagadnieniom dotyczącym wytwarzania nowoczesnych aplikacji internetowych i systemów z wykorzystaniem języka JavaScript i technologii powiązanych. Uczestnik poznaje od podstaw język JavaScript i jego możliwości istotne w tworzeniu nowoczesnych aplikacji webowych. W ramach zajęć uczestnik zdobywa także wiedzę jak korzystać z narzędzi jak Git, npm czy Gulp oraz poznaje platformę Node.js i frameworki Angular i React.js.

Absolwent studiów podyplomowych będzie przygotowany do podjęcia pracy na stanowiskach określanych na rynku pracy jako JavaScript Junior Developer w firmach związanych z wytwarzaniem aplikacji WWW. Będzie posiadał wiedzę i umiejętności z zakresu programowania języka JavaScript, budowy stron internetowych, standardów i technologii WWW oraz pracy w zespole developerskim. Uczestnik zdobywa wiedzę w oparciu o praktyczne przykłady oraz doświadczenie kadry dydaktycznej. Po zakończeniu studiów uczestnik będzie posiadał m.in. wiedzę i umiejętności:

- dotyczące programowania w języku JavaScript, poczynając od podstaw a kończąc na programowaniu obiektowym i zagadnieniach zaawansowanych;
- na temat metodyk pracy przy projektach - Agile i Scrum i kultury pracy deweloperów i tworzenia kodu oraz całego procesu;
- projektowania aplikacji JavaScript z wykorzystaniem wytycznych organizacja kodu, wydajności oraz wzorców projektowych;
- o bieżących standardach kodowania w języku HTML5/CSS3 oraz technikach wytwarzania stron WWW w kontekście aplikacji JavaScript;
- na temat programowania w języku JavaScript wykorzystując do tego popularne frameworki m.in. AngularJS, React;
- projektowania aplikacji ServerSide, programowania i zarządzania systemami z wykorzystaniem środowiska i technologii node.js;
- projektowania i programowania aplikacji mobilnych z wykorzystaniem frameworka wieloplatformowego React Native;
- o testowaniu i walidacji budowanych rozwiązań;
- w posługiwaniu się narzędziami wykorzystywanymi we współczesnej pracy JavaScript Developera oraz praktykami Dev-Ops w wytwarzaniu i utrzymaniu systemów.

Uczestnikiem kursu może być osoba, która posiada kwalifikację pełną co najmniej na poziomie 6 PRK uzyskaną w systemie szkolnictwa wyższego i nauki.

Kandydaci ubiegający się o przyjęcie na kurs powinni mieć podstawową wiedzę i umiejętności z zakresu obsługi komputera i urządzeń peryferyjnych oraz znajomość podstawowych zagadnień związanych z technologiami informacyjnymi, w szczególności związane z obsługą systemów operacyjnych, instalacją oprogramowania, posługiwaniem się konsolą wiersza poleceń MS Windows lub powłoką systemu Linux (shell). Kandydaci nie muszą posiadać wykształcenia technicznego. Niezbędną wiedzę programistyczną dotyczącą języka JavaScript, słuchacz zdobędzie od początku, w trakcie nauki tego przedmiotu na obu semestrach niniejszych studiów. Należy jednak zaznaczyć, że jest to kurs intensywny. Dla osób, które nie miały dotychczas do czynienia z podstawami programowania, udział w zajęciach będzie wymagał dodatkowych nakładów pracy.

Przed rozpoczęciem kursu wymagane jest ukończenie kilku lekcji z kursu JavaScript, np. w serwisach e-learningowych Codecademy, Udemy Pluralsight lub podobnych. Pozytywne zaliczenie tych lekcji będzie podstawą do podjęcia decyzji przez samego uczestnika i definiuje wymagania wstępne dla kierunku studiów. Wymagany zakres to zrozumienie pojęć: zmienne, ciągi znaków, konsola, instrukcje warunkowe i funkcje.

PROGRAM STUDIÓW PODYPLOMOWYCH

JAVASCRIPT DEVELOPER

Studia podyplomowe JavaScript Developer trwają 2 semestry i umożliwiają uzyskanie kwalifikacji cząstkowych na poziomie 6 PRK. Łączna liczba punktów ECTS: 30. Łączna liczba godzin zajęć: 257.

Plan studiów JAVASCRIPT DEVELOPER

Lp.	Nazwa przedmiotu	Kod	Liczba ECTS	Liczba godzin w semestrze			Forma zaliczenia
				W	Ć	PS	
SEMESTR I							
1	Programowanie w języku JavaScript	SPJSD01	3	8	0	18	ZO
2	Angular od podstaw - Programowanie aplikacji Web	SPJSD02	3	8	0	18	ZO
3	Framework i ekosystem React - Wprowadzenie	SPJSD03	3	8	0	18	ZO
4	NodeJS - środowisko i technologia ServerSide 1	SPJSD04	3	8	0	18	ZO
5	Podstawy Agile i Scrum	SPJSD05	2	6	0	6	ZO
6	Metodyka DevOps - produkcja i utrzymanie oprogramowania	SPJSD06	1	3	0	9	ZO
RAZEM W SEMESTRZE			15	41	0	87	Razem godz. 128
SEMESTR II							
1	Zaawansowane zagadnienia języka JavaScript	SPJSD07	3	7	0	21	ZO
2	Zawansowane programowanie aplikacji w Angular	SPJSD08	3	7	0	21	ZO
3	Framework i ekosystem React - Aplikacja serwerowa	SPJSD09	3	7	0	21	ZO
4	NodeJS - środowisko i technologia ServerSide 2	SPJSD10	3	7	0	21	ZO
5	Wytwarzanie oprogramowania z wykorzystaniem Agile i Scrum	SPJSD11	3	8	0	9	ZO
RAZEM W SEMESTRZE			15	36	0	93	Razem godz. 129
RAZEM KIERUNEK [w podziale na FORMY ZAJĘCIOWE]							
ŁĄCZNIE ECTS/GODZ. W TRAKCIE STUDIÓW			30	77	0	180	Razem godz. 257

Zestawienie efektów uczenia się

Zestawienie tabelaryczne kierunkowych efektów uczenia się odnoszących się do charakterystyk drugiego stopnia (poziom 6 PRK), określonych na podstawie Ustawy z dnia 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji.

Objaśnienia oznaczeń:

P6 – poziom 6 PRK (Polska Rama Kwalifikacji)

S – charakterystyka typowa dla kwalifikacji uzyskiwanych w ramach szkolnictwa wyższego

W – wiedza

T – teorie, zasady

Z – zjawiska i procesy

O – organizacja pracy

G – głębia i zakres

K – kontekst

N – narzędzia i materiały

U – umiejętności

I – informacje

W – wykorzystanie wiedzy

K – komunikowanie się

O – organizacja pracy

U – uczenie się i rozwój zawodowy

N – narzędzia i materiały

K – kompetencje społeczne

K – krytyczna ocena

O – odpowiedzialność

R – rola zawodowa

P – przestrzeganie reguł

W – współpraca

JSD – JavaScript Developer

1, 2, 3 i kolejne – numery efektu uczenia się

Załącznik nr 1 do „Wytyczne do tworzenia programów studiów podyplomowych”

Symbol	Efekty Ucznienia się dla kierunku studiów podyplomowych	Odniesienie do charakterystyk drugiego stopnia określonych na podstawie art.7 ust. 3 Ustawy z dn. 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji na poziomie 6 PRK	Odniesienie do charakterystyk drugiego stopnia określonych na podstawie art.7 ust. 4 Ustawy z dn. 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji na poziomie 6 PRK
Wiedza: absolwent zna i rozumie			
JSD_W01	w zaawansowanym stopniu pojęcia i standardy związane z językiem JavaScript, standardem ES6, wzorcami projektowymi i wybranymi frameworkami JavaScript oraz ich zastosowaniem.	P6S_WG	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W02	w zaawansowanym stopniu wybrane konstrukcje, funkcje i zastosowanie języka HTML i w szerokim zakresie języka JavaScript.	P6S_WG	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W03	w zaawansowanym stopniu wybrane współczesne frameworki JavaScript tworzenia wysoce skalowalnych aplikacji internetowych w języku JavaScript w oparciu o paradygmat SOA i microservices.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W04	w zaawansowanym stopniu zasady zarządzania projektem, frameworkiem Scrum, manifestem Agile oraz narzędziami do szacowania prac developerskich.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
JSD_W05	zasady wykorzystywania narzędzi we współczesnej pracy programisty oraz praktyki Dev-Ops w wytwarzaniu i utrzymaniu systemów.	P6S_WG, P6S_WK	P6Z_WT, P6Z_WZ, P6Z_WO
Umiejętności: absolwent potrafi			
JSD_U01	posługiwać się językiem JavaScript zgodnie ze standardem ES6, wzorcami projektowymi i wybranymi frameworkami.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U02	wykorzystać proste konstrukcje HTML oraz język JavaScript w tworzeniu dynamicznych elementów stron internetowych.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U03	budować skalowalne aplikacje internetowe z wykorzystaniem frameworków JavaScript w oparciu o paradygmat SOA i microservices.	P6S_UW, P6S_UK, P6S_UU	P6Z_UI, P6Z_UN, P6Z_UO
JSD_U04	pracować zespołowo, zbudować środowisko pracy i wykorzystywać w pracy wybrane narzędzia procesu Agile i Scrum oraz przygotować dokumentację techniczną.	P6S_UW, P6S_UK, P6S_UO	P6Z_UI, P6Z_UN, P6Z_UO

JSD_U05	wykorzystać narzędzia i brać udział w procesie tworzenia oprogramowania wykorzystując narzędzia i środowisko Dev-Ops.	P6S_UW	P6Z_UI, P6Z_UN, P6Z_UO
Kompetencje społeczne: absolwent jest gotów do			
JSD_K01	rozwiązywania problemów w sposób poznawczy i praktyczny.	P6S_KK	P6Z_KP
JSD_K02	wejścia na rynek pracy jako Junior JavaScript Developer i odpowiedzialnego pełnienia roli zawodowej zgodnie z zasadami etyki oraz dbałością o dorobek zawodu	P6S_KR, P6S_KO	P6Z_KP, P6Z_KW

Zasady oceniania i warunki uzyskania świadectwa ukończenia studiów

Dokumentem stwierdzającym ukończenie studiów podyplomowych jest Świadectwo ukończenia studiów podyplomowych. Wpisuje się do niego ostateczny wynik studiów, którym jest średnia ważona obliczana według wzoru:

$$\text{średnia ocen} = \frac{\sum(\text{ocena z przedmiotu} \times \text{punkty zaliczeniowe ECTS})}{\sum(\text{punktów zaliczeniowych ECTS})}$$

wyrównana do pełnej oceny wg zasady:

- a) do 3,25 – dostateczny,
- b) od 3,26 do 3,75 – dostateczny plus,
- c) od 3,76 do 4,25 – dobry,
- d) od 4,26 do 4,50 – dobry plus,
- e) od 4,51 do 5,00 – bardzo dobry.

Z każdego przedmiotu zostanie wystawiona jedna ocena. Warunkiem uzyskania świadectwa jest uzyskanie wszystkich zaliczeń i egzaminów przewidzianych w planie studiów. Przy zaliczeniach stosuje się następującą skalę ocen:

- a) bardzo dobry – 5,0
- b) dobry plus – 4,5
- c) dobry – 4,0
- d) dostateczny plus – 3,5
- e) dostateczny – 3,0
- f) niedostateczny – 2,0

Nie przewiduje się wykonania pracy końcowej.

Ramowe programy przedmiotów

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Programowanie w języku JavaScript						Kod przedmiotu	SPJSD01	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami programowania JavaScript w standardzie ES6. Uczestnik nabyte umiejętności efektywnego używania podstawowych elementów języka, planowania i tworzenia kodu JS odpowiadającego zadanym wymaganiom oraz dobierania wzorców do zadanego problemu.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Składnia oraz zastosowanie języka JavaScript w standardzie ES6. 2. Nested Scope. 3. Modułowość. 4. Funkcja zwrotne. 5. Organizacja kodu, wydajność oraz wzorce projektowe. <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Składnia oraz zastosowanie języka JavaScript w standardzie ES6. 2. Nested Scope. 3. Modułowość. 4. Funkcja zwrotne. 5. Organizacja kodu, wydajność oraz wzorce projektowe. 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawy programowania JavaScript standard ES6						JSD_W01, JSD_W02		

EU2	Zna metody używania podstawowych elementów języka	JSD_W01, JSD_W02	
EU3	Potrafi zaplanować i stworzyć kod JS odpowiadający zadanym wymaganiom	JSD_U01, JSD_U02	
EU4	Potrafi dobrać wzorzec do zadanego problemu	JSD_U02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	24	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	<ol style="list-style-type: none"> 1. Kirupa Chinnathambi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017 2. Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018 3. Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. JavaScript Tutorial, [02.04.2021] https://www.w3schools.com/JS/default.asp 2. Oficjalna strona dokumentacji Mozilli, [02.04.2021] https://developer.mozilla.org/pl/docs/Web/JavaScript 3. Dokumentacja JavaScript, ES6 przykłady: [02.04.2021] http://es6-features.org/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, mgr inż. Karol Rogowski</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Angular od podstaw - Programowanie aplikacji Web						Kod przedmiotu	SPJSD02	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami programowania w technologii Angular oraz z architekturą aplikacji opartych na komponentach. Uczestnik nabędzie umiejętność wymaganą do tworzenia nowoczesnych aplikacji Web.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Wprowadzenie do technologii Angular. Podstawy języka TypeScript 2. Narzędzia Angular CLI i analiza struktury projektu. Budowa komponentów 3. Wyświetlanie danych w komponencie i dostępne modele komunikacji między komponentami. Style w komponentach – zasady działania. 4. Działanie serwisów logiki aplikacji. Budowanie routingu w aplikacji 5. Implementacja formularzy 6. Podział aplikacji na moduły oraz wdrażanie aplikacji na produkcję <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Podstawy języka TypeScript 2. Narzędzia Angular CLI i analiza struktury projektu 3. Wyświetlanie danych w komponencie i dostępne modele komunikacji między komponentami 4. Działanie serwisów logiki aplikacji 5. Budowanie routingu w aplikacji 6. Implementacja formularzy 7. Podział aplikacji na moduły oraz wdrażanie aplikacji na produkcję 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawy programowania w języku TypeScript						JSD_W02		
EU2	Zna dostępne niezbędne rozwiązania w technologii Angular do						JSD_W02		

	stworzenia aplikacji	
EU3	Potrafi tworzyć aplikacje w architekturze zorientowanej na komponenty	JSD_U02
EU4	Potrafi wykorzystywać możliwości technologii Angular do tworzenia aplikacji Web	JSD_U02
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja
EU1	Test przeprowadzony w systemie e-learningowym	W
EU2	Test przeprowadzony w systemie e-learningowym	W
EU3	Zaliczenie projektem	Ps
EU4	Zaliczenie projektem	Ps
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.
Wyliczenie	Udział w wykładach	8
	Udział w pracowni specjalistycznej	18
	Przygotowanie do pracowni specjalistycznej	10
	Udział w konsultacjach związanych z projektem	5
	Realizacja zadań projektowych (zadań domowych)	34
	RAZEM:	75
Wskaźniki ilościowe		GODZINY ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31 1,24
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		67 2,68
Literatura podstawowa	1. Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017 2. Stephen Radford, Projektowanie nowoczesnych aplikacji sieciowych z użyciem Angular JS i Bootstrapa : poznaj najpopularniejsze frameworki! Gliwice: Helion, 2016. 3. Official Angular Tutorial: [02.04.2021] https://angular.io/tutorial	
Literatura uzupełniająca	1. Angular 4 Tutorial: https://www.tutorialspoint.com/angular4/index.htm 2. Angular: Getting Started: [02.04.2021] https://www.pluralsight.com/courses/angular-2-getting-started-update 3. Angular7 Tutorial – Tutorialspoint: [02.04.2021] https://www.tutorialspoint.com/angular7/index.htm	
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu
Program opracował	<i>dr inż. Paweł Tadejko, Chrystian Ruminowicz</i>	02.04.2021

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Framework i ekosystem React - Wprowadzenie						Kod przedmiotu	SPJSD03	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z frameworkiem React.js. Uczestnik pozna czym jest React.js i dlaczego warto go używać. Uczestnik nabędzie umiejętność budowania dynamicznych aplikacji internetowych.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Środowisko React.js Komponenty - ich struktury, stany i komunikacja między nimi Składnia JSX oraz narzędzia developerskie <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Tworzenie prostej aplikacji internetowej z wykorzystaniem środowiska React.js Podział aplikacji w projekcie na komponenty (i podkomponenty). Budowa wersji statycznej. Zaprojektowanie odwzorowania stanu interfejsu użytkownika. Określenie i implementacja komponentu, który obsługuje zmiany stanów. Implementacja przepływów danych. 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawy oraz strukturę frameworka React.js						JSD_W01, JSD_W02		
EU2	Zna zasady tworzenia komponentów oraz realizacji komunikacji między nimi						JSD_W01, JSD_W02		
EU3	Potrafi stworzyć prostą aplikację						JSD_U02		
EU4	Potrafi używać narzędzi developerskich						JSD_U02		

Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	14	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	30	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	<ol style="list-style-type: none"> 1. Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017 2. Cassio de Sousa Antonio, React dla zaawansowanych, Helion, 2017 3. Adam Freeman, React 16 : framework dla profesjonalistów, Gliwice: Helion, 2020. 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Tutorial: Intro To React, [02.04.2021], https://reactjs.org/tutorial/tutorial.html 2. React Guide, [02.04.2021], https://scotch.io/guides/react 3. Kurs React.js, [02.04.2021], https://typeofweb.com/kurs/react-js/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Konrad Kotelczuk</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	NodeJS - środowisko i technologia ServerSide 1						Kod przedmiotu	SPJSD04	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	8				18			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawami środowiska zaprojektowanego do tworzenia wysoce skalowalnych aplikacji internetowych w języku JavaScript. Uczestnik nabyte umiejętności tworzenia własnego modułu/pakietu.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Środowisko NodeJS - czym jest i jak działa 2. Programowanie asynchroniczne. 3. Podstawy programowania w środowisku NodeJS. 4. Moduł/pakiet – omówienie <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Tworzenie własnego modułu/pakietu. 2. Implementacja aplikacji, która pobiera informacje o użytkowniku i innych danych z serwera i wyświetla w aplikacji. 3. Obsługa żądań asynchronicznych API 4. Zaprojektowania i implementacja podziału na moduły. 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna założenie programowania asynchronicznego						JSD_W01, JSD_W02		
EU2	Zna podstawy programowania w środowisku NodeJS						JSD_W01, JSD_W02, JSD_W05		
EU3	Potrafi korzystać z zewnętrznych modułów/pakietów						JSD_U01, JSD_U02, JSD_U05		
EU4	Potrafi stworzyć własny moduł/pakiet.						JSD_U02, JSD_U05		

Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyczerpanie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	18	
	Przygotowanie do pracowni specjalistycznej	20	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	24	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		31	1,24
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		67	2,68
Literatura podstawowa	<ol style="list-style-type: none"> David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017 Sandro Pasquali, Node.js : projektowanie, wdrażanie i utrzymywanie aplikacji, Gliwice : Helion, 2018. Oficjalna strona dokumentacji: [02.04.2021], https://nodejs.org/en/docs/ 		
Literatura uzupełniająca	<ol style="list-style-type: none"> Jeff Dickey, Nowoczesne aplikacje internetowe: MongoDB, Express, AngularJS, Node.js: poznaj nowe podejście do aplikacji internetowych, Gliwice: Helion, 2016. Brad Dayley, Node.js, MongoDB, AngularJS : kompendium wiedzy, Gliwice : Helion, 2015. 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Paweł Łukaszuk</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Podstawy Agile i Scrum						Kod przedmiotu	SPJSD05	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	6				6			Punkty ECTS	2
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z frameworkiem Scrum, manifestem Agile oraz narzędziami do szacowania prac developerskich. Uczestnik nabędzie umiejętność szacowania prac developerskich.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Główne wytyczne frameworka Agile. Role i odpowiedzialność członków zespołu wynikających z manifestu Agile. Sposoby pracy w zespole Szacowanie prac developerskich. <p>Pracownia specjalistyczna</p> <ol style="list-style-type: none"> Warsztaty pracy w zespole Szacowanie prac developerskich 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: zaliczenie projektem								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna i rozumie zasady manifestu Agile i Frameworka Scrum						JSD_W02, JSD_W04		
EU2	Zna role i obowiązki członków zespołu developerskiego według Frameworka Scrum						JSD_W02, JSD_W04		
EU3	Potrafi wyszacować czasochłonność wyznaczonych zadań						JSD_U02, JSD_U04		
EU4	Potrafi zastosować wytyczne Scrum w pracy w projekcie.						JSD_U02, JSD_U04		
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się						Forma zajęć, na której zachodzi weryfikacja		

EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	6	
	Udział w pracowni specjalistycznej	6	
	Przygotowanie do pracowni specjalistycznej	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	23	
	RAZEM:	50	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		17	0,68
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		44	1,76
Literatura podstawowa	<ol style="list-style-type: none"> Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015 Gunther Verheyen Red. Scrum: 97 rzeczy, które powinieneś wiedzieć : wspólna wiedza wielu ekspertów, Warszawa: Wydaw. Naukowe PWN, 2021. Agile Manifesto, [02.04.2021] http://agilemanifesto.org/ Scrum Guide, Przewodnik, [02.04.2021] http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf 		
Literatura uzupełniająca	<ol style="list-style-type: none"> Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014 Stephanie Ockerman, Simon Reindl, Ken Schwaber, Doskonalenie Scruma: przewodnik dla praktyków : o wyzwaniach, korzyściach i zwinnych zespołach, Gliwice: Helion, 2020. 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Eliza Kasperuk</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Metodyka DevOps - produkcja i utrzymanie oprogramowania						Kod przedmiotu	SPJSD06	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	1
	3				9			Punkty ECTS	1
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z podstawowymi narzędziami wykorzystywanymi w pracy z JavaScriptem tj. systemami kontroli wersji, narzędziami utrzymującymi jakość kodu, narzędziami pozwalającymi na automatyzację zadań. Uczestnik nabędzie umiejętność postawienia swojego serwera developerskiego.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Działanie systemów kontroli wersji. Wprowadzenie do używania GITa Narzędzia automatyzacji pracy. Wprowadzenie do używania Gulp.js Narzędzia utrzymujące jakość kodu. Wprowadzenie do używania ESLint <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Ćwiczenia praktyczne z wykorzystaniem narzędzi używania GITa, Gulp.js, ESLint 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne; 5. dyskusje dydaktyczne.								
Forma zaliczenia	<p>Wykład: test przeprowadzony w systemie e-learningowym</p> <p>Pracownia specjalistyczna: zaliczenie projektem</p>								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawy systemów kontroli wersji						JSD_W03, JSD_W05		
EU2	Zna podstawowe narzędzia automatyzacji pracy						JSD_W03, JSD_W05		
EU3	Potrafi korzystać z GITa						JSD_U03, JSD_U05		
EU4	Potrafi stworzyć serwer developerski i własne skrypty						JSD_U03, JSD_U05		
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się						Forma zajęć, na której zachodzi weryfikacja		
EU1	Test przeprowadzony w systemie e-learningowym						W		
EU2	Test przeprowadzony w systemie e-learningowym						W		

EU3	Zaliczenie projektem	Ps	
EU4	Zaliczenie projektem	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	3	
	Udział w pracowni specjalistycznej	9	
	Przygotowanie do pracowni specjalistycznej	4	
	Udział w konsultacjach związanych z projektem	3	
	Realizacja zadań projektowych (zadań domowych)	6	
	RAZEM:	25	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		15	0,6
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		22	0,88
Literatura podstawowa	<ol style="list-style-type: none"> 1. Jon Loeliger, Matthew McCullough, Kontrola wersji z systemem Git. Narzędzia i techniki programistów, Wydanie II, O'Reilly, 2014 2. Oficjalna dokumentacja GITa: [02.04.2021] https://git-scm.com/docs 3. Oficjalna strona Gulp.js: [02.04.2021] https://gulpjs.com/ 4. Oficjalna dokumentacja Eslint: [02.04.2021] https://eslint.org/docs/user-guide/getting-started 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Gene Kim, Jez Humble, Patrick Debois, John Willis Allspaw Przedm. DevOps : światowej klasy zwinność, niezawodność i bezpieczeństwo w Twojej organizacji, Gliwice : Helion, 2017. 2. Nicole Forsgren, Jez Humble, Gene Kim, Przyspieszenie: Lean i DevOps w rozwoju firm technologicznych, Gliwice, Helion, 2020. 3. Travis Maynard, Getting Started with Gulp - Second Edition, 2017 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował(a)	<i>dr inż. Paweł Tadejko, Kamil Mijacz</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Zaawansowane zagadnienia języka JavaScript						Kod przedmiotu	SPJSD07	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	7				21			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z zaawansowanymi elementami programowania JavaScript w standardzie ES6. Uczestnik nabędzie umiejętność programowanie wielowątkowego, organizacji kodu oraz wykorzystywania wzorców projektowych.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Programowanie wielowątkowe 2. Organizacja kodu 3. Wzorce projektowe i wydajność <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Implementacja wybranych technik wzorców projektowych i programowania wielowątkowego w kodzie programu 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	<p>Wykład: test przeprowadzony w systemie e-learningowym</p> <p>Pracownia specjalistyczna: projekt końcowy</p>								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna zaawansowane elementy programowania JavaScript standardzie ES6						JSD_W01, JSD_W02		
EU2	Zna efektywne metody używania zaawansowanych elementów języka						JSD_W01, JSD_K01		
EU3	Potrafi zaplanować i stworzyć kod JS odpowiadający zadanym wymaganiom przy użyciu wzorców projektowych i wiedzy optymalizacyjnej						JSD_U01, JSD_U03, JSD_K01, JSD_K02		
EU4	Potrafi dobrać wzorzec do zadanego problemu						JSD_U03, JSD_K01, JSD_K02		
Symbol efektu	Sposoby weryfikacji efektów uczenia się						Forma zajęć, na której		

uczenia się		zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	7	
	Udział w pracowni specjalistycznej	21	
	Przygotowanie do pracowni specjalistycznej	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	32	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		33	1,32
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		68	2,72
Literatura podstawowa	<ol style="list-style-type: none"> 1. Kirupa Chinnathambi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017 2. Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018 3. Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. JavaScript Tutorial, [02.04.2021] https://www.w3schools.com/JS/default.asp 2. Oficjalna strona dokumentacji Mozilli, [02.04.2021] https://developer.mozilla.org/pl/docs/Web/JavaScript 3. Dokumentacja JavaScript, ES6 przykłady: [02.04.2021] http://es6-features.org/ 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, mgr inż. Karol Rogowski</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Zawansowane programowanie aplikacji w Angular						Kod przedmiotu	SPJSD08	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	7				21			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest utrwalenie zdobytej wiedzy uczestnika o programowania Angular. Uczestnik nabeździe umiejętność tworzenia w pełni funkcjonalnych aplikacji biznesowych. Uczestnik zdoberdzie umiejętności istotne dla doświadczonego dewelopera, dotyczące utrzymania, rozwoju kodu i pisania testów komponentów.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Proces rozruchu aplikacji 2. Konfiguracja wstrzykiwania zależności 3. Cykl życia komponentu 4. Zaawansowane elementy Angular: dyrektywy i potoki 5. Zaawansowany routing adresów; Komunikacja HTTP z serwerem 6. Testy jednostkowe komponentów; Optymalizacja ładowania zasobów serwera 7. Refactoring aplikacji <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Ćwiczenia programistyczne z wykorzystaniem zaawansowanych elementów Angular: dyrektyw i potoków 2. Ćwiczenia programistyczne z wykorzystaniem zaawansowanego routing adresów 3. Ćwiczenia programistyczne z wykorzystaniem testów jednostkowych komponentów 4. Ćwiczenia programistyczne z wykorzystaniem refactoringu aplikacji 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna zasady działania technologii Angular						JSD_W02, JSD_W03		
EU2	Zna zaawansowane zagadnienia dotyczące logiki komponentów i ich cyklu życia						JSD_W02, JSD_W03		

EU3	Potrafi tworzyć złożone komponenty i pisać do nich testy	JSD_U02, JSD_U03	
EU4	Potrafi komunikować się z serwerem i optymalizować ładowanie zasobów serwera	JSD_U02, JSD_U03	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	7	
	Udział w pracowni specjalistycznej	21	
	Przygotowanie do pracowni specjalistycznej	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	32	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		33	1,32
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		68	2,72
Literatura podstawowa	<ol style="list-style-type: none"> 1. Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017 2. Stephen Radford, Projektowanie nowoczesnych aplikacji sieciowych z użyciem Angular JS i Bootstrapa : poznaj najpopularniejsze frameworki! Gliwice: Helion, 2016. 3. Official Angular Tutorial: [02.04.2021] https://angular.io/tutorial 		
Literatura uzupełniająca	<ol style="list-style-type: none"> 1. Angular 4 Tutorial: https://www.tutorialspoint.com/angular4/index.htm 2. Angular: Getting Started: [02.04.2021] https://www.pluralsight.com/courses/angular-2-getting-started-update 3. Angular7 Tutorial – Tutorialspoint: [02.04.2021] https://www.tutorialspoint.com/angular7/index.htm 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Chrystian Ruminowicz</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Framework i ekosystem React - Aplikacja serwerowa						Kod przedmiotu	SPJSD09	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	7				21			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest rozszerzenie wiedzy uczestnika o podstawach pracy z frameworkiem React.js. Uczestnik nabędzie umiejętność stworzenia aplikacji komunikującej się z serwerem.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Zasady działania routingu Implementacja asynchroniczna komunikacji z API Biblioteki do zarządzania stanami Przepływy danych Store Przedstawienie dobrych praktyk pisania aplikacji Wprowadzenie do testowania komponentów <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Implementacja asynchroniczna komunikacji z API Ćwiczenia programistyczne z zastosowaniem przepływu danych i Store Ćwiczenia programistyczne z zastosowaniem testowania komponentów Stworzenie aplikacji internetowej z wykorzystaniem React.js komunikującej się z API 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna zasady budowania aplikacji z wykorzystaniem routingu						JSD_W02, JSD_W03		
EU2	Zna zasady i ma wiedzę jak zarządzać danymi (stanem aplikacji)						JSD_W02, JSD_W03		
EU3	Potrafi zaimplementować routing i komunikację z API						JSD_U02, JSD_U03		

EU4	Potrafi zarządzać danymi (stanem aplikacji) i testować komponenty	JSD_U02, JSD_U03	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	7	
	Udział w pracowni specjalistycznej	21	
	Przygotowanie do pracowni specjalistycznej	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	32	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		33	1,32
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		68	2,72
Literatura podstawowa	4. Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017 5. Cassio de Sousa Antonio, React dla zaawansowanych, Helion, 2017 6. Adam Freeman, React 16 : framework dla profesjonalistów, Gliwice: Helion, 2020.		
Literatura uzupełniająca	4. Tutorial: Intro To React, [02.04.2021], https://reactjs.org/tutorial/tutorial.html 5. React Guide, [02.04.2021], https://scotch.io/guides/react 6. Kurs React.js, [02.04.2021], https://typeofweb.com/kurs/react-js/		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Konrad Kotelczuk</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	NodeJS - środowisko i technologia ServerSide 2						Kod przedmiotu	SPJSD10	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	7				21			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest rozszerzenie wiedzy uczestnika o podstawach programowania asynchronicznego w środowisku NodeJS. Uczestnik nabeździe umiejętność tworzenia aplikacji sieciowych w środowisku NodeJS.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> 1. Wprowadzenie do protokołów sieciowych 2. Tworzenie aplikacji sieciowych w środowisku NodeJS 3. Praca z wykorzystaniem framework'a Express 4. Stworzenie własnego mikroserwisu 5. Testowanie aplikacji NodeJS <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> 1. Tworzenie aplikacji sieciowych w środowisku NodeJS 2. Ćwiczenia programistyczne z wykorzystaniem framework'a Express 3. Stworzenie własnego mikroserwisu 4. Testowanie aplikacji NodeJS 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	Zna podstawy protokołów sieciowych						JSD_W02, JSD_W03		
EU2	Zna założenie działania aplikacji sieciowych w środowisku NodeJS						JSD_W02, JSD_W03, JSD_W05, JSD_K01		
EU3	Potrafi stworzyć prosty serwer HTTP w oparciu o dostarczone biblioteki						JSD_U02, JSD_U03		

EU4	Potrafi stworzyć testy do aplikacji napisanej w środowisku NodeJS	JSD_U03, JSD_U05, JSD_K02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się	Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym	W	
EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	7	
	Udział w pracowni specjalistycznej	21	
	Przygotowanie do pracowni specjalistycznej	10	
	Udział w konsultacjach związanych z projektem	5	
	Realizacja zadań projektowych (zadań domowych)	32	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		33	1,32
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		68	2,72
Literatura podstawowa	1. David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017 2. Sandro Pasquali, Node.js : projektowanie, wdrażanie i utrzymywanie aplikacji, Gliwice : Helion, 2018. 3. Oficjalna strona dokumentacji: [02.04.2021] https://nodejs.org/en/docs/		
Literatura uzupełniająca	1. Jeff Dickey, Nowoczesne aplikacje internetowe: MongoDB, Express, AngularJS, Node.js: poznaj nowe podejście do aplikacji internetowych, Gliwice: Helion, 2016. 2. Brad Dayley, Node.js, MongoDB, AngularJS : kompendium wiedzy, Gliwice : Helion, 2015.		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Paweł Łukaszuk</i>	02.04.2021	

Wydział Informatyki									
Kierunek studiów	JavaScript Developer						Poziom i forma studiów	Studia podyplomowe	
Specjalność / ścieżka dyplomowania							Profil kształcenia		
Nazwa przedmiotu	Wytwarzanie oprogramowania z wykorzystaniem Agile i Scrum						Kod przedmiotu	SPJSD11	
							Rodzaj przedmiotu	obowiązkowy	
Formy zajęć i liczba godzin	W	C	L	P	Ps	T	S	Semestr	2
	8				9			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie uczestnika z zasadami zarządzania sprintem. Uczestnik nabędzie umiejętność wyceny szacunkowej projektu.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> Sposoby tworzenia ścieżek użytkownika Zarządzania backlogiem produktowym Przygotowanie wyceny projektowej <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> Ćwiczenia z zarządzania backlogiem produktowym Przygotowanie wyceny projektowej 								
Metody dydaktyczne	1. prezentacja multimedialna; 2. analiza wybranych przypadków; 3. samodzielne opracowanie wskazanych zagadnień; 4. ćwiczenia indywidualne, 5. dyskusje dydaktyczne.								
Forma zaliczenia	Wykład: test przeprowadzony w systemie e-learningowym Pracownia specjalistyczna: projekt końcowy								
Symbol efektu uczenia się	Zakładane efekty uczenia się							Odniesienie do kierunkowych efektów uczenia się	
EU1	Zna i rozumie pojęcie sprintu a także jego składowych							JSD_W04, JSD_K01	
EU2	Zna i rozumie zasady zbierania wytycznych projektowych							JSD_W04, JSD_K01	
EU3	Potrafi zaplanować sprint i zmierzyć postęp prac							JSD_U04, JSD_K02	
EU4	Potrafi przygotować szacunkową wycenę projektu							JSD_U04, JSD_K02	
Symbol efektu uczenia się	Sposoby weryfikacji efektów uczenia się							Forma zajęć, na której zachodzi weryfikacja	
EU1	Test przeprowadzony w systemie e-learningowym							W	

EU2	Test przeprowadzony w systemie e-learningowym	W	
EU3	Projekt końcowy	Ps	
EU4	Projekt końcowy	Ps	
Bilans nakładu pracy uczestnika (w godzinach)		Liczba godz.	
Wyliczenie	Udział w wykładach	8	
	Udział w pracowni specjalistycznej	9	
	Przygotowanie do zajęć wykładowych	10	
	Przygotowanie do pracowni specjalistycznej	15	
	Udział w konsultacjach	2	
	Realizacja zadań projektowych (zadań domowych)	31	
	RAZEM:	75	
Wskaźniki ilościowe		GODZINY	ECTS
Nakład pracy uczestnika związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela		19	0,76
Nakład pracy uczestnika związany z zajęciami o charakterze praktycznym		57	2,28
Literatura podstawowa	<ol style="list-style-type: none"> Mariusz Chrapko, Scrum. O zwinnym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015 Gunther Verheyen Red. Scrum: 97 rzeczy, które powinieneś wiedzieć : wspólna wiedza wielu ekspertów, Warszawa: Wydaw. Naukowe PWN, 2021. Agile Manifesto, [02.04.2021] http://agilemanifesto.org/ Scrum Guide, Przewodnik, [02.04.2021] http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf 		
Literatura uzupełniająca	<ol style="list-style-type: none"> Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014 Stephanie Ockerman, Simon Reindl, Ken Schwaber, Doskonalenie Scruma: przewodnik dla praktyków : o wyzwaniach, korzyściach i zwinnych zespołach, Gliwice: Helion, 2020. 		
Jednostka realizująca	Wydział Informatyki Politechniki Białostockiej	Data opracowania programu	
Program opracował	<i>dr inż. Paweł Tadejko, Eliza Kasperuk</i>	02.04.2021	

ZASOBY BIBLIOTECZNE ORAZ ELEKTRONICZNE ZASOBY WIEDZY OBEJMUJĄCE LITERATURĘ ZALECANĄ NA STUDIACH PODYPLOMOWYCH

Biblioteka Politechniki Białostockiej (PB) zapewnia dostęp do zasobów bibliotecznych oraz elektronicznych zasobów wiedzy obejmujących literaturę zalecaną na Studiach Podyplomowych „JavaScript Developer”.

Biblioteka PB jest największą biblioteką naukowo-techniczną w regionie północno-wschodnim Polski. Stanowi podstawę systemu biblioteczo-informacyjnego uczelni, w skład którego wchodzi: Biblioteka Główna, Biblioteka Wydziału Architektury, Biblioteka Wydziału Inżynierii Zarządzania, Biblioteka Instytutu Nauk Leśnych Wydziału Budownictwa i Nauk o Środowisku. Zadaniem Biblioteki PB jest przede wszystkim zaspokajanie potrzeb wszystkich pracowników i studentów w zakresie dostępu do literatury naukowej i dydaktycznej. Biblioteki specjalistyczne obsługują zaś poszczególne wydziały Uczelni, gromadząc i udostępniając księgozbiór ściśle związany z ich potrzebami.

Zasoby biblioteczne, do których uczelnia zapewnia dostęp

1. Adam Freeman, React 16 : framework dla profesjonalistów, Gliwice: Helion, 2020.
2. Brad Dayley, Node.js, MongoDB, AngularJS : kompendium wiedzy, Gliwice : Helion, 2015.
3. Cassio de Sousa Antonio, React dla zaawansowanych, Helion, 2017
4. David Herron, Platforma Node.js Przewodnik webdevelopera, Helion 2017
5. Gene Kim, Jez Humble, Patrick Debois, John Willis Allspaw Przedm. DevOps : światowej klasy zwinność, niezawodność i bezpieczeństwo w Twojej organizacji, Gliwice : Helion, 2017.
6. Gion Kunz, Angular 2. Tworzenie interaktywnych aplikacji internetowych, Helion 2017
7. Gunther Verheyen Red. Scrum: 97 rzeczy, które powinieneś wiedzieć : wspólna wiedza wielu ekspertów, Warszawa: Wydaw. Naukowe PWN, 2021.
8. Jeff Dickey, Nowoczesne aplikacje internetowe: MongoDB, Express, AngularJS, Node.js: poznaj nowe podejście do aplikacji internetowych, Gliwice: Helion, 2016.
9. Jon Loeliger, Matthew McCullough, Kontrola wersji z systemem Git. Narzędzia i techniki programistów, Wydanie II, O'Reilly, 2014
10. Kirupa Chinnambhi. JavaScript. Przewodnik dla absolutnie początkujących, Helion, 2017
11. Luis Atencio. Programowanie funkcyjne z JavaScriptem. Sposoby na lepszy kod, Helion, 2017
12. Mariusz Chrapko, Scrum. O zwinym zarządzaniu projektami, Wydanie II rozszerzone, Helion, 2015
13. Nicolas Bevacqua. Nowoczesny JavaScript. Poznaj ES6 i praktyczne zastosowania nowych rozwiązań, Helion, 2018
14. Nicole Forsgren, Jez Humble, Gene Kim, Przyspieszenie: Lean i DevOps w rozwoju firm technologicznych, Gliwice, Helion, 2020.
15. Roman Pichler: Zarządzanie projektami ze Scrum. Twórz produkty, które pokochają klienci, Helion, 2014
16. Sandro Pasquali, Node.js : projektowanie, wdrażanie i utrzymywanie aplikacji, Gliwice : Helion, 2018.
17. Stephanie Ockerman, Simon Reindl, Ken Schwaber, Doskonalenie Scruma: przewodnik dla praktyków : o wyzwaniach, korzyściach i zwinnych zespołach, Gliwice: Helion, 2020.
18. Stephen Radford, Projektowanie nowoczesnych aplikacji sieciowych z użyciem Angular JS i Bootstrapa : poznaj najpopularniejsze frameworki! Gliwice: Helion, 2016.
19. Stoyan Stefanov, React w działaniu. Tworzenie aplikacji internetowych, Helion, 2017
20. Stoyan Stefanov, React: Up & Running. Building Web Applications (ebook), O'Reilly Media, 2016
21. Travis Maynard, Getting Started with Gulp - Second Edition, 2017

Elektroniczne zasoby wiedzy

1. Agile Manifesto, [02.04.2021] <http://agilemanifesto.org/>
2. Angular 4 Tutorial: [02.04.2021] <https://www.tutorialspoint.com/angular4/index.htm>
3. Angular Best Practices: [02.04.2021] <https://www.pluralsight.com/courses/best-practices-angular>
4. Dokumentacja JavaScript, ES6 przykłady: [02.04.2021] <http://es6-features.org/>
5. JavaScript Tutorial, [02.04.2021] <https://www.w3schools.com/jS/default.asp>
6. Kurs React.js, [02.04.2021] <https://typeofweb.com/kurs/react-js/>
7. NodeJS tutorial: [02.04.2021], <https://www.w3schools.com/nodejs/>
8. Official Angular Tutorial: [02.04.2021] <https://angular.io/tutorial>
9. Oficjalna dokumentacja ESLint: [02.04.2021] <https://eslint.org/docs/user-guide/getting-started>
10. Oficjalna dokumentacja GITa: [02.04.2021] <https://git-scm.com/docs>
11. Oficjalna strona dokumentacji Mozilli, [02.04.2021] <https://developer.mozilla.org/pl/docs/Web/JavaScript>
12. Oficjalna strona dokumentacji: [02.04.2021] <https://nodejs.org/en/docs/>
13. Oficjalna strona Gulp.js: [02.04.2021] <https://gulpjs.com/>
14. React Guide, [02.04.2021] <https://scotch.io/guides/react>

15. Scrum Guide, Przewodnik, [02.04.2021] <http://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-Polish.pdf>
16. Tutorial: Intro To React, [02.04.2021] <https://reactjs.org/tutorial/tutorial.html>