

<b>KARTA OPISU MODUŁU KSZTAŁCENIA</b>		
Nazwa modułu/przedmiotu <b>Frontend Development</b>		Kod <b>1010512331010510135</b>
Kierunek studiów <b>Informatyka</b>	Profil kształcenia (ogólnoakademicki, praktyczny) <b>ogólnoakademicki</b>	Rok / Semestr <b>2 / 3</b>
Ścieżka obieralności/specjalność <b>Gry i technologie internetowe</b>	Przedmiot oferowany w języku: <b>polski</b>	Kurs (obligatoryjny/obieralny) <b>obieralny</b>
Stopień studiów: <b>II stopień</b>	Forma studiów (stacjonarna/niestacjonarna) <b>stacjonarna</b>	
Godziny Wykłady: <b>30</b> Ćwiczenia: - Laboratoria: <b>30</b> Projekty/seminaria: -		Liczba punktów <b>4</b>
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) <b>kierunkowy z danego kierunku</b>		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki <b>nauki techniczne</b> <b>nauki techniczne</b>		Podział ECTS (liczba i %) <b>4 100%</b> <b>4 100%</b>
<b>Odpowiedzialny za przedmiot / wykładowca:</b> dr inż. Marcin Borowski email: mborowski@cs.put.poznan.pl tel. +48 61 665 3032 Wydział Informatyki ul. Piotrowo 3 60-965 Poznań		
<b>Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:</b>		
1	<b>Wiedza:</b>	W zakresie wiedzy, student powinien posiadać podstawową wiedzę z zakresu programowania, technik oraz technologii wykorzystywanych w budowie aplikacji internetowych, sieci komputerowych oraz baz danych.
2	<b>Umiejętności:</b>	Student również powinien posiadać umiejętności rozwiązywania podstawowych problemów z zakresu algorytmiki, optymalizacji kodu programu, pracy z bibliotekami zewnętrznymi, budowy prostych stron internetowych. Wskazany byłaby chęć i dążenie do poszerzania swojej wiedzy i kompetencji.
3	<b>Kompetencje społeczne</b>	Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
<b>Cel przedmiotu:</b>		
1. Przekazanie studentom podstawowej wiedzy dotyczącej technologii wykorzystywanych przy budowie aplikacji webowych w szczególności technik frontendowych, w zakresie podejść do projektowania, doboru technologii oraz implementacji (w tym również rozwiązań przeznaczonych dla urządzeń mobilnych).		
2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z projektowaniem aplikacji internetowych również działających w czasie rzeczywistym (reaktywność), wykorzystywanie frameworków, bibliotek oraz innych narzędzi wspierających budowę serwisów.		
3. Kształtowanie u studentów umiejętności pracy zespołowej jak również samodzielności w rozwiązywaniu problemów.		
<b>Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia</b>		
<b>Wiedza:</b>		
1. ma zaawansowaną i pogłębioną wiedzę z zakresu technologii internetowych, podstaw teoretycznych ich budowania oraz metod, narzędzi i środowisk programistycznych wykorzystywanych do ich implementacji - [K2st_W1]		
2. ma zaawansowaną wiedzę szczegółową dotyczącą technologii frontendowych - [K2st_W3]		
3. ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia aplikacji internetowych oraz przesyłu informacji - [K2st_W5]		
4. zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich podczas budowy aplikacji internetowych - [K2st_W6]		
<b>Umiejętności:</b>		

<p>1. potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji aplikacji frontendowych - [K2st_U2]</p> <p>2. potrafi ? przy formułowaniu i rozwiązywaniu zadań inżynierskich ? integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K2st_U5]</p> <p>3. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć oraz nowych produktów informatycznych (dedykowane narzędzia do transfilacji, deykowane języki itd.) - [K2st_U6]</p> <p>4. potrafi ocenić przydatność metod i narzędzi służących do rozwiązania zadania inżynierskiego, polegającego na budowie lub ocenie systemu informatycznego lub jego składowych, w tym dostrzec ograniczenia tych metod i narzędzi - w efekcie potrafi dobrać odpowiednią technologię wytwarzania aplikacji w zależności od wymagań - [K2st_U9]</p> <p>5. potrafi - zgodnie z zadaną specyfikacją - zaprojektować i zaimplementować złożone aplikacje internetowe - co najmniej w części, używając właściwych metod, technik i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia - [K2st_U11]</p>
<p><b>Kompetencje społeczne:</b></p> <p>1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, w szczególności technologie internetowe - [K2st_K1]</p> <p>2. rozumie potrzeby wykorzystywania najnowszych osiągnięć techniki oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadzić mogą do poważnych strat finansowych, wizerunkowych lub społecznych - [K2st_K2]</p>

<b>Sposoby sprawdzenia efektów kształcenia</b>	
<p>Ocena formująca</p> <p>a. wykład - na podstawie aktywności podczas interaktywnych części wykładów;</p> <p>b. laboratorium - na podstawie oceny bieżącego postępu realizacji zadań;</p> <p>Ocena podsumowująca</p> <p>a. wykład:</p> <ul style="list-style-type: none"> <li>- ocenę samodzielnie przygotowanej przez studenta prezentacji dotyczącej wybranej techniki, biblioteki lub framework'u wykorzystywanego przy budowie aplikacji webowych;</li> <li>- omówienie wyników egzaminu</li> </ul> <p>b. laboratorium - weryfikacja założonych efektów kształcenia realizowanych przez:</p> <ul style="list-style-type: none"> <li>- ocenę i obronę przez studentów przygotowanych zadań - 5 projektów;</li> </ul> <p>Przy wystawianiu oceny końcowej, student może uzyskać podwyższenie oceny za:</p> <ul style="list-style-type: none"> <li>- omówienie dodatkowych aspektów prezentowanych zagadnień, nie prezentowanych na zajęciach;</li> <li>- wykorzystania umiejętności i wiedzy spoza programu studiów do rozwiązywania realizowanych zadań;</li> <li>- pomoc w doskonaleniu materiałów dydaktycznych związanych z przedmiotem;</li> </ul>	
<b>Treści programowe</b>	
<p>Program wykładu obejmuje następujące zagadnienia:</p> <p>Protokół komunikacji HTTP. Wprowadzenie do technologii node.js. Budowa prostych serwerów popularnych usług sieciowych (echo, chat, http). Wprowadzenie do framework'a Express. Wprowadzenie do framework'a MVC Sails.js. Wprowadzenie do framework'a Meteor.js. Omówienie narzędzi wspomagających takich jak Grunt, Bower, SASS, Less, Compass. Języki definiowania szablonów aplikacji EJS, Jade, HAML. Omówienie podejść Angular oraz React.</p> <p>Zajęcia laboratoryjne prowadzone są w formie piętnastu 2-godzinnych ćwiczeń, odbywających się w laboratorium. Ćwiczenia realizowane są samodzielnie przez studentów. Program laboratorium obejmuje następujące zagadnienia:</p> <p>Instalacja i konfiguracja środowiska node.js. Uruchamianie aplikacji napisanych w node.js. Proste serwery usług. Prosta aplikacja TODO zrealizowana w framework'u Express z Angular i MongoDB. Realizacja aplikacji w Sails.js. Realizacja Korzystanie z narzędzi wspomagających oraz modułów dla node.js: Grunt, Forever, Bower, WebPack itp.</p>	
<p><b>Literatura podstawowa:</b></p> <p>1. Dokumentacje techniczne wymienionych narzędzi</p>	
<p><b>Literatura uzupełniająca:</b></p>	
<b>Bilans nakładu pracy przeciętnego studenta</b>	
Czynność	Czas (godz.)

1. udział w wykładach	30	
2. udział w zajęciach laboratoryjnych	30	
3. przygotowanie do zajęć laboratoryjnych	8	
4. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności zajęć laboratoryjnych	2 20	
5. napisanie programów, uruchomienie i testowanie programów (czas poza zajęciami laboratoryjnymi)	10	
6. zapoznanie się ze wskazaną literaturą i materiałami dydaktycznymi (10 str. = 1 godz.), 100 str.	2	
7. przygotowanie prezentacji zaliczeniowej i jej prezentacja		
<b>Obciążenie pracą studenta</b>		
<b>forma aktywności</b>	<b>godzin</b>	<b>ECTS</b>
Łączny nakład pracy	102	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2
Zajęcia o charakterze praktycznym	60	2