

Programowanie zdarzeniowe skalowanych systemów internetowych - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie zdarzeniowe skalowanych systemów internetowych
Kod przedmiotu	11.3-WE-INFP-PZSSI
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2022/2023

Informacje o przedmiocie	
Semestr	6
Liczba punktów ECTS do zdobycia	5
Typ przedmiotu	obieralny
Język nauczania	polski
Sylabus opracował	- dr inż. Jacek Tkacz - dr inż. Michał Doligalski

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	15	1	9	0,6	Zaliczenie na ocenę
Laboratorium	30	2	18	1,2	Zaliczenie na ocenę
Projekt	15	1	9	0,6	Zaliczenie na ocenę

Cel przedmiotu

Zapoznanie z metodami tworzenia i rozwoju aplikacji internetowych na pojedynczej stronie z wykorzystaniem szkieletów projektowych JavaScript oraz ich praktyczne wykorzystanie.

Zapoznanie z mechanizmami programowania asynchronicznego. Wykorzystanie wzorców projektowych w projektowaniu skalowalnych aplikacji internetowych.

Wymagania wstępne

- Język Java i technologie Web
- Programowanie współbieżne i rozproszone
- Zarządzanie przemysłowym projektem informatycznym

Zakres tematyczny

Tworzenie i rozwój aplikacji internetowych w językach JavaScript//TypeScript na pojedynczej stronie z wykorzystaniem wzorców projektowych. Wiązanie danych w Angular.js oraz React.js. Statyczne i dynamiczne źródła danych JSON. Node.js jako środowisko uruchomieniowe zaprojektowane do tworzenia wysoce skalowalnych aplikacji internetowych. Obsługa zdarzeń w Node.js. Tworzenie aplikacji sterowanych zdarzeniami, wykorzystujących asynchroniczny system wejścia-wyjścia z wykorzystaniem mechanizmów async, await oraz promises (obietnic). Techniki tworzenia Web-API (Rest Api/GraphQL) w Node.js. Dwukierunkowa komunikacja między przeglądarką a serwerem w oparciu o Websockets.

Metody kształcenia

Wykład: wykład konwencjonalny i multimedialny.

Laboratorium: ćwiczenia laboratoryjne

Projekt: Wykonanie zadań projektowych, praca w zespole

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Studen ma wiedze niezbędną z zakresu projektowania i implementacji systemów reaktywnych oraz tworzenia asynchronicznych aplikacji internetowych.	K_U15	kolokwium	Wykład

Opis efektu	Symbol e efektów	Metody weryfikacji	Forma zajęć
Ma umiejętności z zakresu wykorzystania frameworków i nowoczesnych narzędzi programistycznych do tworzenia reaktywnych aplikacji interenowych.	• K_W19 • K_U20	• wykonanie sprawozdań laboratoryjnych	• Laboratorium
Student jest zdolny kreatywnie zaplanować i zrealizować projekt informatyczny realizujący aplikacje internetowe.	• K_U20 • K_K05 • K_K09	• projekt • wykonanie sprawozdań laboratoryjnych	• Laboratorium • Projekt

Warunki zaliczenia

Wykład - Warunkiem zaliczenia jest uzyskanie pozytywnej oceny z kolokwium realizowanego w formie pisemnej.

Laboratorium - warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych, przewidzianych do realizacji w ramach programu laboratorium (80%) oraz aktywności na zajęciach (20%).

Projekt - warunkiem zaliczenia jest uzyskanie pozytywnej oceny z zadania projektowego, oraz ocen cząstkowych.

Składowe oceny końcowej = wykład: 40% + laboratorium: 30% + projekt: 30%

Literatura podstawowa

1. Herron D., Platforma Node.js. Przewodnik webdeveloper – trzecie wydanie, Helion, Warszawa, 2017.
2. Fernando Doglio. REST API development with Node.JS., 2018.
3. Mardan A., Practical Node.js Building Real-World Scalable Web Apps, Apress, Berkeley, 2018.
4. Sandro Pasquali. Node.js. Projektowanie, wdrażanie i utrzymywanie aplikacji. Helion, 2017.
5. Mike Cantelon. Node.js in Action, Second Edition. Manning Publications, 2017.
6. Andrew Mead. Learning Node.js Development. 2018.
7. Stoyan Stefanov. React w działaniu. Tworzenie aplikacji internetowych. Helion, 2017.
8. David Choi. React, TypeScript i Node. Tworzenie aplikacji internetowych typu fullstack. Helion, 2022.
9. Freeman A.: AngularJS. Profesjonalne techniki, Helion, 2015.
10. Fain Y., Moiseev A.: Angular 2. Programowanie z użyciem języka TypeScript, Helion, 2017
11. Kalbarczyk D., Kalbarczyk A.: AngularJS. Pierwsze kroki, Helion, 2015

Literatura uzupełniająca

Uwagi

Zmodyfikowane przez dr inż. Jacek Tkacz (ostatnia modyfikacja: 08-04-2022 13:34)

Wygenerowano automatycznie z systemu SyllabUZ