

Modern design of internet applications - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Modern desig of internet applications
Kod przedmiotu	11.3-WE-INF-D-MDoIA-Er
Wydział	Wydział Informatyki, Elektrotechniki i Automatyki
Kierunek	Informatyka
Profil	ogólnoakademicki
Rodzaj studiów	Program Erasmus
Semestr rozpoczęcia	semestr zimowy 2017/2018

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	6
Typ przedmiotu	obieralny
Język nauczania	angielski
Sylabus opracował	<ul style="list-style-type: none">• dr hab. inż. Remigiusz Wiśniewski, prof. UZ• dr inż. Grzegorz Bazydło• dr inż. Iwona Grobelna

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	-	-	Zaliczenie na ocenę
Laboratorium	30	2	-	-	Zaliczenie na ocenę
Projekt	15	1	-	-	Zaliczenie na ocenę

Cel przedmiotu

- Familiarize students with the designing of Internet applications using MVC pattern, modern script languages and frameworks.
- Familiarize students with practical aspects of scripting languages for solving complex problems.

Wymagania wstępne

Zakres tematyczny

Design of interactive Internet services using markup languages (*HTML*, *XHTML*, *HTML5*) and style sheets *CSS* – correct structure and hierarchy of a service (navigation, interactivity, usability), block elements, embedded hyperlinks, tables, forms.

Validation of *HTML* and *CSS* code. Web design. Design of Internet applications usage *MVC* pattern (Model-View-Controller). Usage of script languages for Internet services. Usage of *PHP* for server applications, data structures, application design, *GET* and *POST* methods, cookies.

Security of Internet applications based on script languages (e.g. *PHP*). Threads related to cookies, data passing (*POST* and *GET* method). Protection against code injections (Cross-Site Scripting, *SQL-Injection*, etc.).

Python: usage – console applications and Internet services, complex structures, data and files, script implementation in console (data protection, scripts for daily work), applications in *Python* for embedded systems (e.g. for Raspberry Pi), access to databases.

Development of services and Internet applications using frameworks, e.g. *Symfony*, *ZendFramework*, *Django*.

Metody kształcenia

Lecture, laboratory exercises, project.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Knows future trends for that discipline and understand the needs for using modern techniques, methods and tools supporting design of Internet applications.		<ul style="list-style-type: none">• bieżąca kontrola na zajęciach• kolokwium	<ul style="list-style-type: none">• Wykład

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Knows existing solutions, can analyze them and use some design methods to solve a complex scientific and engineering problem.		<ul style="list-style-type: none"> bieżąca kontrola na zajęciach projekt 	<ul style="list-style-type: none"> Laboratorium Projekt
Can choose appropriate tools and techniques to solve complex engineering problems related to design of digital systems.		<ul style="list-style-type: none"> projekt 	<ul style="list-style-type: none"> Projekt
Can use modern tools and design methods for web applications to realize some part of a bigger project or system. Can cooperate and work in a team.		<ul style="list-style-type: none"> bieżąca kontrola na zajęciach dyskusja obserwacje i ocena umiejętności praktycznych studenta projekt sprawdzian 	<ul style="list-style-type: none"> Laboratorium Projekt

Warunki zaliczenia

Lecture – the passing condition is to obtain a positive mark from the final test.

Laboratory – the passing condition is to obtain positive marks from all laboratory exercises to be planned during the semester.

Project – the passing condition is to obtain a positive mark from all projects conducted during the semester.

Final mark components: lecture 30% + laboratory 40% + project 30%.

Literatura podstawowa

1. M. Lutz, Python. Wprowadzenie. Wydanie III, Helion, Gliwice, 2009.
2. J. Forcier, P. Bissex, W. Chun, Python i Django. Programowanie aplikacji webowych, Helion, Gliwice, 2009.
3. E. Freeman, E. Freeman, Head First HTML with CSS & XHTML. Edycja polska (Rusz głową!), Helion, Gliwice, 2007.
4. E. Watrall, J. Siarto, Head First Web Design. Edycja polska, Helion, Gliwice, 2010.

Literatura uzupełniająca

1. W. Gajda, Symfony 2 od podstaw, Helion, Gliwice, 2012.
2. G. Halfacree, E. Upton, Raspberry Pi. Przewodnik użytkownika, Helion, Gliwice, 2013.
3. T. Skaraczyński, A. Zoła, PHP5. Programowanie z wykorzystaniem Symfony, CakePHP, Zend Framework, Helion, Gliwice, 2009.
4. W. Gajda, Zend Framework od podstaw. Wykorzystaj gotowe rozwiązania PHP do tworzenia zaawansowanych aplikacji internetowych, Helion, Gliwice, 2011.
5. E. T. Freeman, E. Robson, HTML5. Rusz głową!, Helion, Gliwice, 2012.
6. L. Welling, L. Thomson, PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty. Wydanie trzecie, Helion, Gliwice, 2005.
7. C. Pitt, Wzorzec MVC w PHP dla profesjonalistów, Helion, Gliwice, 2013.

Uwagi

Zmodyfikowane przez dr hab. inż. Remigiusz Wiśniewski, prof. UZ (ostatnia modyfikacja: 14-05-2017 02:12)

Wygenerowano automatycznie z systemu SylabUZ