

Programowanie aplikacji internetowych - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie aplikacji internetowych
Kod przedmiotu	13.2-WI-GeoTSP-Progr.aplik.-S18
Wydział	Wydział Budownictwa, Architektury i Inżynierii Środowiska
Kierunek	Geoinformatyka i techniki satelitarne
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2019/2020

Informacje o przedmiocie	
Semestr	4
Liczba punktów ECTS do zdobycia	4
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	

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
Laboratorium	60	4	36	2,4	Zaliczenie na ocenę

Cel przedmiotu

Celem przedmiotu jest wprowadzenie do programowania aplikacji internetowych zarówno w warstwie prezentacji danych (programowanie po stronie przeglądarki) jak i w warstwie analizy, gromadzenia i przetwarzania danych (programowanie po stronie serwera z wykorzystaniem baz danych). Oprócz rozwijania umiejętności programowania aplikacji internetowych, celem przedmiotu jest również rozwijanie świadomości wolnego oprogramowania i umiejętności wykorzystywania go w tworzeniu profesjonalnych aplikacji internetowych.

Wymagania wstępne

Znajomość podstaw programowanie w języku Python z uwzględnieniem programowania obiektowego, znajomość podstaw relacyjnych baz danych. Znajomość języka angielskiego na poziomie zapewniającym swobodne rozumienie treści pisanych - dokumentacji technicznych oraz użytkowych oraz popularnych artykułów traktujących o technologii.

Zakres tematyczny

1. HTML

- struktura dokumentu HTML i jej model obiektowy (DOM)
- elementy blokowe i „liniowe”
- prezentacja treści (dobre praktyki)
- praca z grafiką
- odnośniki i nawigacja pomiędzy dokumentami
- prezentacja informacji tabelarycznych i paginacja
- formularze i wysyłanie danych do serwera
- optymalizacja SEO

2. CSS

- selektory, model pudełkowy i pozycjonowanie
- podstawy stylizacji różnych treści (tekst, grafiki, tabele, formularze, etc.)
- złożone dokumenty CSS - import, scalanie i kompresja
- frameworki CSS na przykładzie Bootstrap
- preprocesory CSS na przykładzie SASS

3. JavaScript

- podstawy języka JavaScript
- asynchroniczne ładowanie treści (AJAX)
- ważne frameworki JS na przykładzie JQuery i React
- graficzna prezentacja danych i grafiki interaktywne

4. Aplikacje internetowe na przykładzie Django

- koncepcja dynamicznego dostarczania treści

- podstawy konfiguracji sieciowej i rozproszenia usług (serwer HTTP, serwer bazodanowy, WSGI i aplikacja)
- serwery pośredniczące i równoważenie obciążenia (proxy i load-balancing)
- mapowanie obiektowo-relacyjne (ORM): SQL i modelowanie zarządzania informacją
- przechowywanie i przetwarzanie danych po stronie serwera (*backend*)
- prezentacja dynamicznie ładowanych treści (*frontend*) - języki szablonów
- architektura REST i AJAX oraz współpraca z JavaScript
- uwierzytelnianie, zabezpieczanie dostępu do treści i szyfrowanie komunikacji
- obsługa błędów i komunikacja z użytkownikiem końcowym

Metody kształcenia

Ćwiczenia laboratoryjne, metoda projektu, praca w grupie, giełda pomysłów, burza mózgów, prezentacja, praca z dokumentacją, samodzielne pozyskiwanie wiedzy.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Student potrafi stworzyć aplikację internetową spełniającą standardy HTML/CSS zgodnie ze standardami W3C, potrafi oddzielić warstwę danych od warstwy prezentacji i omówić konieczność takiego podziału.	• K_U07 • K_U09	• bieżąca kontrola na zajęciach • dyskusja • projekt	• Laboratorium
Student potrafi dobrać i skorzystać z gotowych bibliotek, frameworków i rozwiązań bez naruszania cudzej własności intelektualnej i z zachowaniem zapisów licencji. Potrafi wskazać wolne oprogramowanie stanowiące profesjonalną alternatywę dla oprogramowania komercyjnego. Jest świadomy roli wolnego oprogramowania w rozwoju aplikacji internetowych i jego znaczenia na rynku pracy	• K_W04 • K_W05 • K_U01	• bieżąca kontrola na zajęciach • dyskusja • projekt	• Laboratorium
Student potrafi zaprojektować serwis internetowy prezentujący dynamicznie dostarczane treści generowane w oparciu o ich bazodanowy model. Student potrafi zaimplementować opracowany projekt używając frameworka Django. Potrafi omówić rolę bazodanowych serwisów internetowych oraz jest świadomy ich znaczenia na współczesnych rynkach pracy	• K_W04 • K_W09 • K_U07	• bieżąca kontrola na zajęciach • dyskusja • projekt	• Laboratorium

Warunki zaliczenia

Ocena końcowa jest oceną wykonania oraz omówienia projektu końcowego wybranego z listy zaproponowanych projektów lub własnej propozycji przyjętej przez osobę prowadzącą. W projekcie końcowym oceniane jest właściwe użycie HTML + CSS + JS (40% oceny), zaprojektowanie modeli, widoków i zaimplementowanie całości serwisu w frameworku Django (40% oceny), omówienie projektu i technik w nim zastosowanych (20% oceny).

Literatura podstawowa

- https://developer.mozilla.org/pl/
- https://docs.djangoproject.com

Literatura uzupełniająca

Uwagi

Zmodyfikowane przez dr inż. Gerard Bryś (ostatnia modyfikacja: 29-04-2019 09:58)

Wygenerowano automatycznie z systemu SylabUZ