



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Programowanie aplikacji internetowych [N2Inf1-ZTI>PAI]

Przedmiot

Kierunek studiów
Informatyka

Rok/Semestr
1/1

Studia w zakresie (specjalność)
Zaawansowane technologie internetowe

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
polski

Forma studiów
niestacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
16

Laboratorium
16

Inne
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr inż. Marcin Borowski
marcin.borowski@put.poznan.pl

Wykładowcy

dr inż. Marcin Borowski
marcin.borowski@put.poznan.pl
dr inż. Andrzej Urbański
andrzej.urbanski@put.poznan.pl

Wymagania wstępne

W zakresie wiedzy, student powinien posiadać podstawową wiedzę z zakresu programowania strukturalnego oraz obiektowego, podstawową wiedzę na temat technologii internetowych oraz podstawową wiedzę z zakresu projektowania i używania baz danych. Powinien posiadać umiejętność rozwiązywania podstawowych problemów związanych z procesem projektowania systemów informatycznych oraz umiejętność pozyskiwania informacji ze wskazanych źródeł. W zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.

Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy dotyczącej projektowania aplikacji internetowych oraz mobilnych, przepływu informacji w takich aplikacjach, technologii wykorzystywanych przy ich budowie (w tym również aplikacji na urządzenia mobilne). 2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z projektowaniem aplikacji internetowych i mobilnych, wykorzystywania rozwiązań typu Open Source, framework"ów oraz bibliotek wspomagających budowę tego typu rozwiązań. 3. Kształtowanie u studentów umiejętności pracy zespołowej oraz samodzielności w rozwiązywaniu problemów.

Przedmiotowe efekty uczenia się

Wiedza:

Student:

- ma uporządkowaną, podbudowaną teoretycznie wiedzę w zakresie technologii sieciowych i aplikacji internetowych mających związek z budową aplikacji internetowych
- ma szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki wykorzystywanych przy budowie aplikacji internetowych
- ma wiedzę o trendach rozwojowych w informatyce i w wybranych pokrewnych dyscyplinach - technologie wykorzystywane do budowy aplikacji internetowych
- ma wiedzę o cyklu życia aplikacji internetowych, cyklu życia danych przetwarzanych w aplikacjach internetowych
- zna zaawansowane metody i techniki stosowane przy realizacji zadań inżynierskich związanych z budową aplikacji internetowych

Umiejętności:

Student:

- potrafi przy formułowaniu i rozwiązywaniu zadań inżynierskich integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) jak również wiedzę z obszaru działania danej aplikacji internetowej oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne
- potrafi pozyskiwać informacje na temat aplikacji internetowych z literatury, baz danych oraz innych źródeł (w języku polskim i angielskim), integrować je i dokonywać ich interpretacji i oceny, wyciągać wnioski i formułować opinie
- potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć techniki (metod, narzędzi, bibliotek, framework"ów, usług) oraz nowych produktów informatycznych
- potrafi określać kierunki dalszego uczenia się i zdobywać nową wiedzę na temat aplikacji internetowych i przekazywać ją innym

Kompetencje społeczne:

Student:

- rozumie potrzeby wykorzystywania najnowszych osiągnięć techniki oraz zna przykłady i rozumie przyczyny wadliwie działających aplikacji, które doprowadzić mogą do poważnych strat finansowych, wizerunkowych lub społecznych
- rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, w szczególności technologie internetowe i mobilne
- potrafi ustalać priorytety służące realizacji określonego przez siebie lub innych zadania w szczególności podczas realizacji projektów informatycznych

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca

- a. wykład - na podstawie aktywności podczas interaktywnych części wykładów;
- b. laboratorium - na podstawie oceny bieżącego postępu realizacji zadań;

Ocena podsumowująca

a. wykład

- ocenę zdobytej wiedzy oraz umiejętności wykazanych na egzaminie - teście przeprowadzonym przy użyciu aplikacji internetowej; pytania o charakterze ogólnym oraz problemowym (student może korzystać z dowolnych materiałów dydaktycznych; test wyboru z ok. 30 pytaniami; zaliczenie wg następujących kryteriów uzyskanego rezultatu:

- > 50% - 3.0
- > 60% - 3.5
- > 70% - 4.0
- > 80% - 4.5
- > 90% - 5.0

• omówienie wyników egzaminu

b. laboratorium

- weryfikacja założonych efektów kształcenia realizowanych przez:

- ocenę i obronę przez studentów przygotowanych zadań - 5 projektów;

Przy wystawianiu oceny końcowej, student może uzyskać podwyższenie oceny za:

- omówienie dodatkowych aspektów prezentowanych zagadnień, nie prezentowanych na zajęciach;

- wykorzystania umiejętności i wiedzy spoza programu studiów do rozwiązywania realizowanych zadań;

- pomoc w doskonaleniu materiałów dydaktycznych związanych z przedmiotem;

Treści programowe

Wykład:

Program wykładu obejmuje następujące zagadnienia:

Strukturę oraz składnię języka HTML5, CSS3, JavaScript. Podstawy node.js, express.js oraz kilku aktualnych narzędzi frontend'owych.

Laboratorium:

Zajęcia laboratoryjne prowadzone są w formie szesnastu godzinnych zajęć odbywających się w laboratorium komputerowym. Pierwsze zajęcia przeznaczone są na zapoznanie studentów z zasadami użytkowania laboratorium i zaliczania ćwiczeń.

Program zajęć laboratoryjnych obejmuje następujące zagadnienia:

Tworzenie stron w HTML5. Tworzenie arkuszy styli CSS3, LESS, SASS. Podstawy języka

JavaScript oraz bibliotek pomocniczych. Budowa aplikacji z wykorzystaniem omawianych, nowoczesnych technologiach frontend'owych.

Tematyka zajęć

Wykład:

- Struktura i składnia HTML5, CSS3, JavaScript.

- Podstawy Node.js i Express.js.

- Aktualne narzędzia frontendowe.

Laboratorium:

- Tworzenie stron w HTML5.

- Tworzenie arkuszy styli CSS3, LESS, SASS.

- Podstawy JavaScript i bibliotek pomocniczych.

- Budowa aplikacji z nowoczesnymi technologiami frontendowymi.

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.

Ćwiczenia laboratoryjne: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, live coding oraz wykonanie zadań podanych przez prowadzącego - ćwiczenia praktyczne.

Literatura

- Bootstrap Framework – <http://getbootstrap.com>

- TailwindCSS - <https://tailwindcss.com/>

- Javascript - <https://www.javascript.com/>

- Svelte - <https://svelte.dev/>

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	34	1,50
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwίων/egzaminu, wykonanie projektu)	66	2,50