



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Frontend Development

### Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Inżynieria oprogramowania

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

2/3

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

angielski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

30

Laboratoria

30

### Liczba punktów ECTS

5

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Marcin Borowski

email: marcin.borowski@cs.put.poznan.pl

tel: 61 665 3032

wydział: Wydział Informatyki i Telekomunikacji

adres: ul. Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Marcin Borowski

email: mborowski@cs.put.poznan.pl

tel: +48 61 665 30 32

wydział: Wydział Informatyki i Telekomunikacji

adres: Piotrowo 2, Poznań

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z programowania strukturalnego oraz obiektowego, programowania z wykorzystaniem schematu MVC, podstawowej wiedzy na temat technologii internetowych (HTML, CSS, JS), oraz podstawową wiedzę z zakresu projektowania baz danych.

Powinien posiadać umiejętność rozwiązywania podstawowych problemów związanych z procesem projektowania systemów informatycznych oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.



Powinien również rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto 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 technologii wykorzystywanych przy budowie aplikacji webowych w szczególności technik frontend'owych, 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 framework'ów, bibliotek oraz innych narzędzi wspierających budowę serwisów.
3. Kształtowanie u studentów umiejętności pracy zespołowej jak również samodzielności w rozwiązywaniu problemów.

### **Przedmiotowe**

### **efekty**

### **uczenia**

### **się**

#### Wiedza

- ma zaawansowaną i pogłębioną wiedzę z zakresu technologii internetowych, podstaw teoretycznych ich budowania oraz metod, narzędzi i środowisk programistycznych wykorzystywanych do ich implementacji
- ma zaawansowaną wiedzę szczegółową dotyczącą technologii frontend'owych ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia aplikacji internetowych oraz przesyłu informacji
- zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich podczas budowy aplikacji internetowych

#### Umiejętności

- potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć oraz nowych produktów informatycznych (dedykowane narzędzia, dedykowane języki itd.)
- potrafi — przy formułowaniu i rozwiązywaniu zadań inżynierskich — integrować wiedzę z różnych obszarów informatyki oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne



- potrafi posługiwać się technikami informacyjno-komunikacyjnymi wykorzystywanymi przy realizacji aplikacji frontend'owych
- 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
- 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

#### Kompetencje społeczne

- rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe, w szczególności technologie internetowe
- 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

#### 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 samodzielnie przygotowanej przez studenta prezentacji dotyczącej wybranej techniki, biblioteki lub framework'u wykorzystywanego przy budowie aplikacji webowych;
- 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: Protokół komunikacji HTTP. Wprowadzenie do technologii node.js. Budowa prostych serwerów popularnych usług sieciowych (echo, chat, http). Wprowadzenie do framework'a Express.js. Wprowadzenie do framework'a AngularJS. Wprowadzenie do framework'a ReactJS. Wprowadzenie do framework'a Meteor.js. Wprowadzenie do biblioteki Svelte. Omówienie narzędzi wspomagających takich jak Grunt, Gulp, Webpack, Rollup, SASS, Less, Postcss. Języki definiowania szablonów aplikacji i komponentów EJS, Jade, HAML, JSX.

### Laboratorium:

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: Przygotowanie szablonów stron i widoków komponentów z wykorzystaniem HTML5, CSS, LESS, SASS oraz wykorzystania framework'ów i bibliotek komponentów ([m.in.](#) Bootstrap, SemanticUI, Tailwindcss). Instalacja i konfiguracja środowiska node.js. Uruchamianie aplikacji napisanych w node.js. Proste serwery usług. Realizacja prostych aplikacji zrealizowane w framework'u Express.js z AngularJS i MongoDB, ReactJS, Meteor.js, Svelte/Sapper. Przykłady wykorzystania narzędzi wspomagających oraz modułów dla node.js: Gulp, Webpack, Nodemon, Rollup itp.

## 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

Podstawowa

1. Dokumentacje techniczne wymienionych narzędzi dostępne w internecie

Uzupełniająca



**Bilans nakładu pracy przeciętnego studenta**

	Godzin	ECTS
Łączny nakład pracy	125	5,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,5
Praca własna studenta (studia literaturowe; przygotowanie do zajęć laboratoryjnych; udział w konsultacjach; przygotowanie programów, uruchomienie i testowanie; przygotowanie prezentacji zaliczeniowej i jej prezentacja)	65	2,5