



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Aplikacje internetowe [S2ETI2>AI]

Przedmiot

Kierunek studiów

Edukacja techniczno-informatyczna

Rok/Semestr

1/1

Studia w zakresie (specjalność)

–

Profil studiów

ogólnoakademicki

Poziom studiów

drugiego stopnia

Język oferowanego przedmiotu

polski

Forma studiów

stacjonarne

Wymagalność

obligatoryjny

Liczba godzin

Wykład

30

Laboratorium

30

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

Wymagania wstępne

Student powinien posiadać podstawową wiedzę z zakresu programowania strukturalnego i obiektowego, technologii internetowych, projektowania oraz użytkowania baz danych. Niezbędne są umiejętności rozwiązywania problemów projektowych oraz korzystania z literatury i dokumentacji technicznej. W zakresie kompetencji społecznych oczekuje się rzetelności, odpowiedzialności, kreatywności i kultury osobistej.

Cel przedmiotu

1. Zapoznanie studentów z zasadami projektowania aplikacji internetowych, przepływem danych i stosowanymi technologiami. 2. Kształcenie umiejętności tworzenia rozwiązań opartych na frameworkach i bibliotekach open source. 3. Doskonalenie pracy zespołowej, samodzielności i umiejętności planowania działań projektowych.

Przedmiotowe efekty uczenia się

Wiedza:

1. Posiada wiedzę o metodach, technikach i narzędziach inżynierskich stosowanych przy projektowaniu oraz implementacji aplikacji internetowych, w tym frameworków, bibliotek i wzorców projektowych

wykorzystywanych w środowisku sieciowym

2. Zna zasady projektowania architektury aplikacji webowych w ujęciu systemowym, w tym komponentów frontendowych i backendowych, oraz ich integracji z bazami danych
3. Ma uporządkowaną i pogłębioną wiedzę o językach programowania webowego, zasadach działania protokołów HTTP, technologii klient-serwer oraz bezpieczeństwa aplikacji sieciowych
4. Zna aktualne trendy i kierunki rozwoju współczesnych technologii internetowych, w tym zastosowania frameworków open source oraz narzędzi do automatyzacji i testowania aplikacji

Umiejętności:

1. Potrafi przeanalizować wymagania użytkownika i zaprojektować złożoną aplikację internetową, wybierając adekwatne technologie i narzędzia programistyczne (np. HTML5, CSS3, JavaScript, Tailwindcss, Sveltekit)
2. Potrafi pozyskiwać i interpretować informacje z dokumentacji technicznej, repozytoriów kodu oraz źródeł branżowych w języku polskim i angielskim w celu doskonalenia tworzonych rozwiązań
3. Umie zastosować zaawansowane techniki informatyczne do budowy i testowania aplikacji webowych, w tym narzędzia do automatyzacji, wersjonowania i integracji środowisk (np. Git, Vite, NPM/Bun)
4. Potrafi zaplanować i przeprowadzić proces testowania, wdrażania oraz dokumentowania tworzonej aplikacji, analizować błędy i wprowadzać usprawnienia w kodzie źródłowym
5. Potrafi skutecznie komunikować się w zespole projektowym, prezentując rozwiązania problemów technicznych i uzasadnienia projektowe w sposób zrozumiały dla odbiorcy technicznego i nietechnicznego
6. Potrafi kierować małym zespołem projektowym realizującym aplikację internetową oraz rozdzielać zadania zgodnie z kompetencjami członków zespołu

Kompetencje społeczne:

1. Jest gotowy do samodzielnego aktualizowania wiedzy z zakresu nowych technologii internetowych i rozwoju zawodowego w obszarze inżynierii oprogramowania
2. Jest przygotowany do współpracy w zespołach projektowych, otwarty na konstruktywną krytykę i gotowy do dzielenia się wiedzą i doświadczeniem
3. Wykazuje odpowiedzialność za jakość i bezpieczeństwo opracowanych aplikacji, przestrzega zasad etyki zawodowej oraz ochrony praw autorskich i własności intelektualnej

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

1. w zakresie wykładów:
 - a. na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,
2. w zakresie laboratoriów / ćwiczeń:
 - a. na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez:

1. ocenę wiedzy i umiejętności wykazanych w kolokwium o charakterze problemowym (student może korzystać z dowolnych materiałów dydaktycznych) / w formie testu wyboru, (30 pytań za łączną liczbę 60 punktów (2 punkty za prawidłową odpowiedź), zaliczenie od 60%+1)

• omówienie wyników kolokwium,

b) w zakresie laboratoriów / ćwiczeń weryfikowanie założonych efektów kształcenia realizowane jest przez:

1. ocenę i „obronę” przez studenta sprawozdania z realizacji projektu,

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

1. omówienia dodatkowych aspektów zagadnienia,

• efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,

1. uwagi związane z udoskonaleniem materiałów dydaktycznych,

wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Kryteria oceny /ocena: zgodnie z regulaminem studiów

Treści programowe

Program zajęć obejmuje naukę podstaw technologii webowych, w tym HTML5, CSS3 oraz JavaScript, wraz z wykorzystaniem nowoczesnych bibliotek i frameworków, takich jak React czy SvelteKit. Studenci poznają zasady tworzenia stron internetowych i aplikacji webowych, a także podstawy programowania w JS/TS.

Tematyka zajęć

Program wykładu obejmuje następujące zagadnienia:

Podstawy technologii webowych. Strukturę oraz składnię języka HTML5, CSS3. Podstawy języka JavaScript, wykorzystanie biblioteki JS. Budowanie aplikacji z wykorzystaniem React/SvelteKit.

Ćwiczenia laboratoryjne prowadzone są w formie piętnastu dwugodzinnych zajęć odbywających się w laboratorium komputerowym. Pierwsze zajęcia przeznaczone są na zapoznanie studentów z zasadami użytkowania laboratorium i zaliczania ćwiczeń. Ćwiczenia realizowane są przez dwuosobowe zespoły studentów. Program zajęć laboratoryjnych obejmuje następujące zagadnienia:

Tworzenie stron w HTML5. Tworzenie arkuszy stylu CSS3. Wykorzystywanie bibliotek JavaScript. Programowanie w języku JS/TS. Budowa aplikacji z wykorzystaniem nowoczesnych framework'ów i bibliotek.

Metody dydaktyczne

1. wykład: prezentacja multimedialna, prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań, pokaz multimedialny, demonstracja.
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, dyskusja, praca w zespole, pokaz multimedialny, studium przypadków, demonstracja.

Literatura

Podstawowa:

1. HTML5 - <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. CSS3 - <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. Tailwindcss Framework - <https://tailwindcss.com/>
4. SvelteKit Documentation - <http://svelte.dev>

Uzupełniająca:

-

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	62	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwium/egzaminu, wykonanie projektu)	38	2,00