

SYLABUS ZAJĘĆ /GRUPY ZAJĘĆ

Dane ogólne:

| | | | | | |
|-----------------------------|---------------------------------|----------------|---------------|-------------------|------|
| Jednostka organizacyjna: | Katedra Informatyki | | | | |
| Kierunek studiów: | Informatyka | | | | |
| Specjalność /Specjalizacja: | Inżynieria oprogramowania | | | | |
| Nazwa zajęć / grupy zajęć : | Technologie aplikacji webowych | | | | |
| Course / group of courses: | Web Application Technologies | | | | |
| Forma studiów: | stacjonarne | | | | |
| Nazwa katalogu: | WP-IN-I-20/21Z-IO | | | | |
| Nazwa bloku zajęć : | | | | | |
| Kod zajęć /grupy zajęć : | 105915 | Kod Erasmus: | | | |
| Punkty ECTS: | 3 | Rodzaj zajęć : | | obowiązkowy | |
| Rok studiów: | 3 | Semestr: | | 6 | |
| Rok | Semestr | Forma zajęć | Liczba godzin | Forma zaliczenia | ECTS |
| 3 | 6 | LO | 30 | Zaliczenie z ocen | 2 |
| | | W | 15 | Egzamin | 1 |
| Razem | | | 45 | | 3 |
| Koordynator: | magister inżynier Dariusz Piwko | | | | |
| Prowadzący zajęcia: | | | | | |
| Język wykładowy: | semestr: 6 - język polski | | | | |

Objaśnienia:

Rodzaj zajęć : obowiązkowe, do wyboru.

Forma prowadzenia zajęć : W - wykład, - wiczenia audytoryjne, L - lektorat, S – seminarium/ zajęcia seminaryjne, P - wiczenia praktyczne (w tym zajęcia wf), M - wiczenia specjalistyczne (medyczne/ kliniczne), LO – wiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zajęcia z technologii informacyjnych, P – wiczenia projektowe, ZT – zajęcia terenowe, T - wiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

Dane merytoryczne

| Wymagania wstępne: | | | |
|---|--|---------------------------------|---|
| Znajomość podstaw programowania oraz terminologii z nim związanej. Sprawne posługiwanie się elementarnymi instrukcjami programistycznymi. | | | |
| Szczegółowe efekty uczenia się | | | |
| Lp. | Student, który zaliczył zajęcia zna i rozumie/potrafi/jest gotowy do: | Kod efektu dla kierunku studiów | Sposób weryfikacji efektu uczenia się |
| 1 | Umie praktycznie zastosować poznane technologie, biblioteki oraz frameworki do implementacji aplikacji webowej. | IN1_W04 | ocena aktywności |
| 2 | Zna pojęcie interfejsu aplikacji oraz użytkownika. Potrafi dobrać odpowiedni interfejs według potrzeb aplikacji. | IN1_W07 | obserwacja wykonania zadania, ocena aktywności |
| 3 | Potrafi zastosować technologie komponentowe, obiektowo relacyjne, modułowe, a także grafowe w implementacji aplikacji WWW. | IN1_W07 | obserwacja wykonania zadania, kolokwium, ocena aktywności |
| 4 | Posiada wiedzę na temat stosowanych technologii, jej doboru do potrzeb oraz integracji z innymi systemami w programowaniu aplikacji WWW. | IN1_W08 | kolokwium, ocena aktywności |

| | | | |
|--|---|------------------|---|
| 5 | Potrafi zaprojektować aplikację WWW w wybranej technologii. | IN1_W08, IN1_W04 | kolokwium, ocena aktywności |
| 6 | Potrafi zastosować technologie komponentowe, obiektowo relacyjne, modułowe, a także grafowe w implementacji aplikacji WWW. | IN1_U01 | obserwacja wykonania zadania, kolokwium, ocena aktywności |
| 7 | Umie praktycznie zastosować poznane technologie, biblioteki oraz frameworki do implementacji aplikacji webowej. | IN1_U05 | ocena aktywności |
| 8 | Zna pojęcie interfejsu aplikacji oraz użytkownika. Potrafi dobrać odpowiedni interfejs według potrzeb aplikacji. | IN1_U12, IN1_U14 | obserwacja wykonania zadania, ocena aktywności |
| Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne) | | | |
| metody podaje (Przedstawione treści na wykładzie w oparciu o prezentację multimedialną oraz analizę oraz omawiania praktycznych przykładów.), metody praktyczne (ćwiczenia laboratoryjne prowadzone na podstawie instrukcji (udostępnianych w sposób elektroniczny) z przykładami i zadaniami do samodzielnej realizacji, które podlegają ocenie.) | | | |
| Kryteria oceny i weryfikacji efektów uczenia się | | | |
| wiedza: ocena kolokwium (ocena kolokwium) obserwacja wykonania zadania (obserwacja) ocena aktywności (ocena aktywności na zajęciach) umiejętności: ocena kolokwium (ocena kolokwium) obserwacja wykonania zadania (obserwacja) ocena aktywności (ocena aktywności na zajęciach) | | | |
| Warunki zaliczenia | | | |
| Kolokwium przeprowadzone na laboratorium oceniane jest w poniższej skali (zakreślenie na korzyść studenta) Skala ocen: 5.0 (bdb) 100 - 91% 4.5 (+db) 90% - 81% 4.0 (db) 80% - 71% 3.5 (+dst) 70% - 61% 3.0 (dst) 60% - 51% 2.0 (ndst) <50% Ocena końcowa jest to średnia ocen z kolokwium przy zachowaniu w/w skali. | | | |
| Treści programowe (opis skrócony) | | | |
| 1. Projektowanie interfejsów użytkownika. 2. Podstawy protokołu HTTP/S. Omówienie kodów zapytań. 3. Formaty wymiany danych. 4. Autentykacja i autoryzacja w aplikacjach webowych. 5. Integracja z bazami danych. 6. Konteneryzacja środowiska pracy. 7. Preprocesory CSS. 8. Pojęcie mikroservisu. 9. Tworzenie dostępnych i użytecznych aplikacji webowych. 10. Narzędzia programistyczne. | | | |
| Content of the study programme (short version) | | | |
| 1. Designing user interfaces. 2. Basics of HTTP / S. Overview of query codes. 3. Data exchange formats. 4. Authentication and authorization in web applications. 5. Integration with databases. 6. Containerization of the work environment. 7. CSS preprocessors. 8. The concept of microservice. 9. Creating accessible and useful web applications. 10. Development tools. | | | |
| Treści programowe | | | |
| | | | Liczba godzin |
| Semestr: 6 | | | |
| Forma zajęć: wykład | | | |
| 1. | Projektowanie interfejsów użytkownika (np. API, REST API, GraphQL) na przykładzie różnych technologii (np. NodeJS, Python). | 15 | |
| 2. | Podstawy protokołu HTTP/S. Omówienie kodów zapytań. Zabezpieczenie błędów aplikacji i | | |

| | |
|---|----|
| <p>wysyłanie przyjaznych komunikatów błędów dla użytkowników.</p> <ol style="list-style-type: none"> 3. Formaty wymiany danych w aplikacjach webowych (np. XML, JSON) 4. Autentykacja i autoryzacja aplikacji webowych (np. JWT, OAuth). Hashowanie hasła. 5. Integracja aplikacji webowej z bazami danych w celu przechowywania danych przesłanych przez urządzenia zewnętrzne (np. PostgreSQL, MongoDB). Dobieranie bazy według potrzeb aplikacji. 6. Konteneryzacja środowiska pracy przy użyciu dostępnych narzędzi (np. Docker). Pojęcie obrazu, konteneru. 7. Preprocesory CSS. Poznanie składni Sass oraz Less. 8. Pojęcie mikroserwisu. Komunikacja aplikacji (różnych technologii) i wymiana danych pomiędzy nimi. 9. Tworzenie i projektowanie dostępnych oraz użytecznych aplikacji webowych. Stosowanie wskazówek organizacji zajmujących się tworzeniem specyfikacji (np. W3C, WAI-ARIA). Pojęcie wykluczenia cyfrowego. 10. Korzystanie z narzędzi programistycznych, ułatwiających pracę przy implementacji aplikacji webowych. 11. Przegląd innych technologii webowych (Java Spring, Ruby on Rails). | 15 |
| Forma zajęć : wiczenia laboratoryjne | |
| <ol style="list-style-type: none"> 1. Projektowanie interfejsów użytkownika (np. API, REST API, GraphQL) na przykładzie różnych technologii (np. NodeJS, Python). 2. Podstawy protokołu HTTP/S. Omówienie kodów zapytań. Zabezpieczenie błędów aplikacji i wysyłanie przyjaznych komunikatów błędów dla użytkowników. 3. Formaty wymiany danych w aplikacjach webowych (np. XML, JSON) 4. Autentykacja i autoryzacja aplikacji webowych (np. JWT, OAuth). Hashowanie hasła. 5. Integracja aplikacji webowej z bazami danych w celu przechowywania danych przesłanych przez urządzenia zewnętrzne (np. PostgreSQL, MongoDB). Dobieranie bazy według potrzeb aplikacji. 6. Konteneryzacja środowiska pracy przy użyciu dostępnych narzędzi (np. Docker). Pojęcie obrazu, konteneru. 7. Preprocesory CSS. Poznanie składni Sass oraz Less. 8. Pojęcie mikroserwisu. Komunikacja aplikacji (różnych technologii) i wymiana danych pomiędzy nimi. 9. Tworzenie i projektowanie dostępnych oraz użytecznych aplikacji webowych. Stosowanie wskazówek organizacji zajmujących się tworzeniem specyfikacji (np. W3C, WAI-ARIA). Pojęcie wykluczenia cyfrowego. 10. Korzystanie z narzędzi programistycznych, ułatwiających pracę przy implementacji aplikacji webowych. 11. Przegląd innych technologii webowych (Java Spring, Ruby on Rails). | 30 |
| Literatura | |
| Podstawowa | |
| Bezpieczeństwo aplikacji webowych, Michał Sajdak (red.), Securitum Szkolenia, Kraków 2019 | |
| Chris Aquino, Todd Gande, Podręcznik frontendowca. The Big Nerd Ranch Guide, Helion 2017 | |
| Christopher Schmitt, Kyle Simpson, HTML5 Cookbook, Helion 2011 | |
| Peter Gasston, CSS3. Podręcznik nowoczesnego webdevelopera, Helion 2015 | |
| Sean P. Kane, Karl Matthias, Docker. Praktyczne zastosowania. Wydanie II, Helion 2019 | |
| Dokumentacja techniczna dostępna w sieci Internet opisująca użyte frameworki podczas zajęć. | |
| Uzupełniająca | |

Dane jako ciowe

| Przyporzkowanie zaj /grup zaj do dyscypliny naukowej/artystycznej | | informatyka techniczna i telekomunikacja | |
|---|--|--|------|
| Sposób okre lenia liczby punktów ECTS | | | |
| Forma nakładu pracy studenta (udział w zaj ciach, aktywno , przygotowanie sprawozdania, itp.) | | Obci enia studenta [w godz.] | |
| Udział w zaj ciach | | 45 | |
| Konsultacje z prowadz cym | | 0 | |
| Udział w egzaminie | | 0 | |
| Bezpo redni kontakt z nauczycielem - inne | | 0 | |
| Przygotowanie do laboratorium, wicze , zaj | | 10 | |
| Przygotowanie do kolokwiiów i egzaminu | | 10 | |
| Indywidualna praca własna studenta z literatur , wykładami itp. | | 5 | |
| Inne | | 5 | |
| Sumaryczne obci enie prac studenta | | 75 | |
| Liczba punktów ECTS | | | |
| Liczba punktów ECTS | | 3 | |
| Zaj cia wymagaj ce bezpo redniego udziału nauczyciela akademickiego | | L. godzin | ECTS |
| | | 45 | 1,8 |
| Zaj cia o charakterze praktycznym | | L. godzin | ECTS |
| | | 45 | 1,8 |

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zaj wymagaj cych bezpo redniego udziału nauczyciela akademickiego i o charakterze praktycznym mo e si ró ni od ł cznej liczby punktów ECTS dla zaj /grup zaj .