

SYLABUS ZAJ /GRUPY ZAJ

Dane ogólne:

| | | | | | |
|----------------------------|---------------------------------|--------------|---------------|-------------------|------|
| Jednostka organizacyjna: | Katedra Informatyki | | | | |
| Kierunek studiów: | Informatyka | | | | |
| Specjalno /Specjalizacja: | | | | | |
| Nazwa zaj / grupy zaj : | Programowanie reaktywne | | | | |
| Course / group of courses: | Reactive Programming | | | | |
| Forma studiów: | stacjonarne | | | | |
| Nazwa katalogu: | WP-IN-I-20/21Z | | | | |
| Nazwa bloku zaj : | | | | | |
| Kod zaj /grupy zaj : | 105998 | Kod Erasmus: | | | |
| Punkty ECTS: | 4 | Rodzaj zaj : | | fakultatywny | |
| Rok studiów: | 2 | Semestr: | | 4 | |
| Rok | Semestr | Forma zaj | Liczba godzin | Forma zaliczenia | ECTS |
| 2 | 4 | LO | 30 | Zaliczenie z ocen | 3 |
| | | W | 15 | Zaliczenie z ocen | 1 |
| Razem | | | 45 | | 4 |
| Koordinator: | magister in ynier Dariusz Piwko | | | | |
| Prowadz cy zaj cia: | | | | | |
| J zyk wykładowy: | semestr: 4 - j zyk polski | | | | |

Obja nienia:

Rodzaj zaj : obowi zkowe, do wyboru.

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

Dane merytoryczne

| Wymagania wst pne: | | | |
|--|---|---------------------------------|---|
| Znajomo podstaw programowania oraz terminologii z nim zwi zanej. Sprawne posługiwanie si elementarnymi instrukcjami programistycznymi. | | | |
| Szczegółowe efekty uczenia si | | | |
| Lp. | Student, który zaliczył zaj cia zna i rozumie/potrafi/jest gotowy do: | Kod efektu dla kierunku studiów | Sposób weryfikacji efektu uczenia si |
| 1 | Posiada wiedz na temat stosowanych technologii w programowaniu aplikacji WWW. | IN1_W04 | kolokwium, ocena aktywno ci |
| 2 | Posiada usystematyzowan podstawow wiedz , która pozwoli na poznanie nowszych technologii | IN1_W04 | obserwacja wykonania zada , ocena aktywno ci |
| 3 | Potrafi praktycznie zastosowa poznane technologie do implementacji aplikacji WWW. Ma wiadomo cyklu ycia projektu oraz jego projektowania. | IN1_W08, IN1_W04 | kolokwium, ocena aktywno ci |
| 4 | Potrafi korzysta z dokumentacji technicznej standardowych pakietów/bibliotek oraz komponentów. | IN1_U01, IN1_U12 | obserwacja wykonania zada , kolokwium, ocena aktywno ci |

| | | | |
|---|---|---------|--|
| 5 | Potrafi zaprojektować aplikację WWW w oparciu o wzorzec projektowy MVC. | IN1_U02 | kolokwium, ocena aktywności |
| 6 | Potrafi wykorzystać dotychczasową wiedzę do łatwego przyswojenia nowych technologii | IN1_U12 | obserwacja wykonania zadania, ocena aktywności |
| 7 | Posiada usystematyzowaną podstawową wiedzę, która pozwoli na poznanie nowszych technologii | IN1_U14 | obserwacja wykonania zadania, ocena aktywności |
| 8 | Potrafi współpracować w rozproszonym zespole nad wspólnym kodem. | IN1_K05 | ocena aktywności |
| Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne) | | | |
| metody podające (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 bezpośrednia studenta w czasie wykonywania działań właściwych dla danego zadania zawodowego wynikających z ról) ocena aktywności (ocena aktywności na zajęciach) umiejętności: ocena kolokwium (ocena kolokwium) obserwacja wykonania zadania (obserwacja bezpośrednia studenta w czasie wykonywania działań właściwych dla danego zadania zawodowego wynikających z ról) ocena aktywności (ocena aktywności na zajęciach) kompetencje społeczne: ocena aktywności (ocena aktywności na zajęciach) | | | |
| Warunki zaliczenia | | | |
| Wykład: Zaliczenie na podstawie obecności. Laboratorium: Zaliczenie z oceną wystawioną na podstawie pozytywnego zaliczenia kolokwium z zadania, aktywności na zajęciach (w rozwiązywaniu zadań i problemów). Ocena końcowa jest średnią ważoną ocen częściowych uzyskanych z w/w zadań. Zaliczanie zajęć jest oceniane zgodnie ze skalą ocen określoną w Regulaminie Studiów PWSZ. | | | |
| Treści programowe (opis skrócony) | | | |
| 1. Podstawy wersjonowania kodu na przykładzie aplikacji internetowej. 2. Tworzenie aplikacji internetowych – dobre praktyki. 3. Skryptowe języki programowania i ich biblioteki z wykorzystaniem programowania reaktywnego. 4. Przegląd technologii internetowych, zasady projektowania aplikacji reaktywnych, modele aplikacji. 5. Zarządzanie stanem aplikacji. 6. Zagadnienia dostępności oraz użyteczności aplikacji i stron internetowych. Korzystanie ze specyfikacji i dokumentacji technicznej. 7. Tworzenie wyizolowanych komponentów, zarządzających swoim własnym stanem. 8. Bezpieczeństwo danych aplikacji internetowych. 9. Praca z mockup'ami. | | | |
| Content of the study programme (short version) | | | |
| 1. Basics of code versioning, for example, an internet application. 2. Creating web applications - good practices. 3. Script programming languages and their libraries using reactive programming. 4. Review of internet technologies, rules for designing reactive applications, application models. 5. Application state management. 6. Accessibility and usability issues of applications and websites. Use of specifications and technical documentation. 7. Creating isolated components that manage their own state. 8. Data security of internet applications. 9. Work with mockups. | | | |
| Treści programowe | | | |
| | | | Liczba godzin |
| Semestr: 4 | | | |
| Forma zajęć : wykład | | | |
| 1. | Podstawy wersjonowania kodu (GIT). | 15 | |
| 2. | Przegląd technologii aplikacji internetowych oraz zasady tworzenia i projektowania aplikacji. | | |
| 3. | Wprowadzenie do technologii HTML oraz CSS. | | |

| | |
|---|----|
| 4. Języki skryptowe (np. JavaScript). 5. Przegląd środowisk oraz frameworków do tworzenia aplikacji internetowych (np. ReactJS, NodeJS). 6. Zarządzanie stanem aplikacji na przykładzie zewnętrznych bibliotek służących do tworzenia interfejsów użytkownika (np. ReactJS) 7. Zasady responsywności, stylizacja aplikacji. Zastosowanie dodatkowych bibliotek i frameworków (np. Bootstrap, Material Design) 8. Podstawy protokołu HTTP/S. Omówienie kodów zapytań 9. Zasady tworzenia aplikacji użytkowej i dostępnej. Przedstawienie specyfikacji ze wskazówkami tworzenia stron internetowych (np. W3C, WAI-ARIA). 10. Udoskonalanie aplikacji webowej zgodnie z zasadami User Experience. 11. Bezpieczeństwo aplikacji webowych zagrażające bezpieczeństwu danych 12. Praca z mockup'ami. Tworzenie uniwersalnych interfejsów użytkownika. 13. Narzędzia ułatwiające tworzenie aplikacji webowych | 15 |
|---|----|

Forma zajęć : **wiczenia laboratoryjne**

| | |
|---|----|
| 1. Podstawy wersjonowania kodu (GIT). 2. Przegląd technologii aplikacji internetowych oraz zasady tworzenia i projektowania aplikacji. 3. Wprowadzenie do technologii HTML oraz CSS. 4. Języki skryptowe (np. JavaScript). 5. Przegląd środowisk oraz frameworków do tworzenia aplikacji internetowych (np. ReactJS, NodeJS). 6. Zarządzanie stanem aplikacji na przykładzie zewnętrznych bibliotek służących do tworzenia interfejsów użytkownika (np. ReactJS) 7. Zasady responsywności, stylizacja aplikacji. Zastosowanie dodatkowych bibliotek i frameworków (np. Bootstrap, Material Design) 8. Podstawy protokołu HTTP/S. Omówienie kodów zapytań 9. Zasady tworzenia aplikacji użytkowej i dostępnej. Przedstawienie specyfikacji ze wskazówkami tworzenia stron internetowych (np. W3C, WAI-ARIA). 10. Udoskonalanie aplikacji webowej zgodnie z zasadami User Experience. 11. Bezpieczeństwo aplikacji webowych zagrażające bezpieczeństwu danych 12. Praca z mockup'ami. Tworzenie uniwersalnych interfejsów użytkownika. 13. Narzędzia ułatwiające tworzenie aplikacji webowych | 30 |
|---|----|

Literatura

Podstawowa

Chris Aquino, Todd Gandee, Podręcznik frontendowca. The Big Nerd Ranch Guide, Helion 2017

Christopher Schmitt, Kyle Simpson, HTML5 Cookbook, Helion 2011

Kirupa Chinnathambi, React i Redux. Praktyczne tworzenie aplikacji WWW, Addison-Wesley Professional 2018

Michał Miszczyszyn, TypeScript na poważnie, Type of Web, Gdańsk 2020

Peter Gasston, CSS3. Podręcznik nowoczesnego webdevelopera, Helion 2015

Dokumentacja techniczna dostępna w sieci Internet opisująca użyte frameworki podczas zajęć

Uzupełniająco

Dane jako ciowe

| | |
|--|---|
| Przyporządkowanie zajęć/grup zajęć do dyscypliny naukowej/artystycznej | informatyka techniczna i telekomunikacja |
| Sposób określania liczby punktów ECTS | |
| Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.) | Obciążenie studenta [w godz.] |
| Udział w zajęciach | 45 |

| | | |
|---|------------------|-------------|
| Konsultacje z prowadzącym | 2 | |
| Udział w egzaminie | 0 | |
| Bezporedni kontakt z nauczycielem - inne | 3 | |
| Przygotowanie do laboratorium, wicze, zaj | 20 | |
| Przygotowanie do kolokwium i egzaminu | 15 | |
| Indywidualna praca własna studenta z literatury, wykładami itp. | 5 | |
| Inne | 10 | |
| Sumaryczne obciążenie prac studenta | 100 | |
| Liczba punktów ECTS | | |
| Liczba punktów ECTS | 4 | |
| Zajęcia wymagające bezporedniego udziału nauczyciela akademickiego | L. godzin | ECTS |
| | 50 | 2,0 |
| Zajęcia o charakterze praktycznym | L. godzin | ECTS |
| | 78 | 3,1 |

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

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zajęć wymagających bezporedniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć /grup zajęć.