



**POLITECHNIKA
BYDGOSKA**

Wydział Telekomunikacji,
Informatyki i Elektrotechniki

Karta przedmiotu
Programowanie aplikacji sieciowych

1. Informacje podstawowe

Kierunek studiów teleinformatyka Jednostka zarządzająca kierunkiem studiów Wydział Telekomunikacji, Informatyki i Elektrotechniki Poziom studiów pierwszego stopnia (inż.) Profil studiów Profil ogólnoakademicki Forma studiów studia niestacjonarne		Cykl kształcenia (nabór) 2024/25 Kod przedmiotu 05TINN.PI18C.1215.24 Języki wykładowe polski Obligatoryjność Obowiązkowy Blok zajęciowy Przedmioty kierunkowe	
Wymagania wstępne	Podstawy systemów operacyjnych		
Przedmioty wprowadzające	Programowanie obiektowe		
Koordinator	Piotr Grad		
Okres Semestr 4	Forma zaliczenia Zaliczenie na ocenę Forma prowadzenia i godziny zajęć Wykład: 9		Liczba punktów ECTS 1.0
Okres Semestr 5	Forma zaliczenia Zaliczenie na ocenę Forma prowadzenia i godziny zajęć Ćwiczenia laboratoryjne: 18		Liczba punktów ECTS 3.0

2. Efekty uczenia się dla przedmiotu

Kod	Opis efektów uczenia się	Odniesienie do kierunkowych efektów uczenia się	Odniesienie do charakterystyk PRK
Wiedza:			
W1	Student ma wiedzę w zakresie architektury systemów operacyjnych i sieci komputerowych, rozumie zasady działania protokołów sieciowych w stopniu umożliwiającym tworzenie aplikacji sieciowych.	TIN_O1_K_W09	P6S_WG P6S_WG_inż
W2	Student ma wiedzę w zakresie działania urządzeń wchodzących w skład sieci teleinformatycznych, w stopniu umożliwiającym tworzenie aplikacji sieciowych.	TIN_O1_K_W11	P6S_WG P6S_WG_inż
W3	Student ma wiedzę dotyczącą współczesnych technik bezpieczeństwa systemów teleinformatycznych, umożliwiającą odpowiednie zabezpieczenie tworzonych aplikacji sieciowych.	TIN_O1_K_W16	P6S_WG P6S_WG_inż
Umiejętności:			
U1	Student posiada umiejętności umożliwiające korzystanie z literatury fachowej (również w języku obcym).	TIN_O1_K_U01	P6S_UW P6S_UW_inż
U2	Student posiada umiejętności umożliwiające korzystanie z bibliotek programistycznych ułatwiających tworzenie aplikacji sieciowych.	TIN_O1_K_U09	P6S_UW P6S_UW_inż
Kompetencje społeczne:			
K1	Student rozumie znaczenie przedsiębiorczości, innowacyjności i kreatywności w wykonywaniu zadań.	TIN_O1_K_K05	P6S_KO

3. Treści programowe

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
1.	1. Przegląd technologii programistycznych wykorzystujących HTML5, CSS3, JS, PHP, MySQL/MariaDB. 2. Przegląd wybranych protokołów i usług sieciowych wykorzystywanych przy programowaniu aplikacji sieciowych. 3. Podstawy wykorzystania środowiska programistycznego twórcy aplikacji webowych, desktopowych i mobilnych: Visual Studio 2022 Community oraz Visual Studio Code. Przegląd narzędzi deweloperskich, ich modułów, rozszerzeń, wersji. 4. Podstawowe funkcjonalności komercyjnych serwisów hostingowych i porównanie jakości serwisów najpopularniejszych usługodawców funkcjonujących w Polsce i na świecie. 5. Przegląd architektury tworzonych aplikacji sieciowych typu klient-serwer. Cienki i gruby klient. 6. Podstawy programowania aplikacji sieciowych w C# na platformę Windows 11, Android oraz iOS z wykorzystaniem serwera MySQL/MariaDB dostępnego w ramach wybranego hostingu komercyjnego. 7. Budowanie aplikacji przeznaczonych jednocześnie na przeglądarkę internetową oraz na system Windows 11, Android oraz iOS. 8. Aplikacje sieciowe budowane z wykorzystaniem sterowników Arduino Ethernet oraz Raspberry Pi 9. Bezpieczeństwo aplikacji sieciowych. Narzędzia i zalecenia OWASP	Wykład	W1, W2, W3

Lp.	Treści programowe	Formy zajęć	Efekty uczenia się dla przedmiotu
2.	1. W ramach ćwiczeń laboratoryjnych studenci uczą się praktycznej implementacji zagadnień omawianych na wykładzie, tzn. tworzenia aplikacji webowych w języku PHP i JS w środowisku Visual Studio Code oraz aplikacji desktopowych i mobilnych z wykorzystaniem środowiska Visual Studio 2022 Community. 2. Przygotowanie środowiska pracy – wykupienie komercyjnego hostingu lub skorzystanie z bezpłatnego hostingu, instalacja VSC oraz narzędzi webmasterskich oraz VS2022. 3. Proces logowania w aplikacjach webowych. Audyt bezpieczeństwa aplikacji sieciowych (SQL Injection, Brute Force), wykorzystanie zaleceń oraz narzędzi OWASP. Audyt bezpieczeństwa starszych dystrybucji WordPress’a. Zgłaszanie prośby o odblokowanie hostingu w przypadku wykrycia źródła ataków przez systemy IDS dostawcy usług hostingowych. 4. Bazedanowa aplikacja sieciowa do monitorowania wybranych usług w sieciach IP. 5. Aplikacja do rejestrowania rozszerzonych informacji o gościach portalu internetowego oraz monitorowania usług działających na hostingu. Wykorzystanie usługi Google Analytics w tworzonych aplikacjach webowych. 6. Komunikator internetowy oraz podsłuchiwanie aplikacji webowych. Tworzenie aplikacji mobilnych umożliwiających wymianę informacji z komunikatorami internetowymi. 7. Internetowa aplikacja typu SCADA oparta na grafice wektorowej oraz wybranych elementach technologii IoT, wykorzystująca technologie: AJAX oraz SSE. 8. Aplikacja typu Customer Relationship Management, budowana przy wykorzystaniu wybranych narzędzi wspierających budowanie i zarządzanie baz danych np. MySQL Workbench. 9. Aplikacja typu prywatna chmura oraz jej dokumentacja przy wykorzystaniu portalu github.com. 10. CMS wielojęzycznego portalu firmowego z „inteligentnym asystentem” typu chatterbot, zbudowanym przy wykorzystaniu języka AIML. 11. Aplikacja e-learningowa zawierająca kursy oraz testy wielokrotnego wyboru z wybranej dziedziny oraz generowanie raportów z przeprowadzonych testów w formacie PDF. 12. Internetowe forum dyskusyjne z zarządzaniem wątkami oraz automatycznym blokowaniem postów zawierających niecenzuralne wyrażenia. 13. Aplikacja sieciowa o tematyce wybranej przez studenta i ustalonej na początku semestru. Aplikacja może być realizowana przez dwójkę studentów, aby przygotować ich do konieczności współpracy, przy realizacji większych projektów.	Ćwiczenia laboratoryjne	U1, U2, K1

4. Metody prowadzenia zajęć, weryfikacji efektów uczenia się i warunki zaliczenia

Semestr 4

Forma zajęć	
-------------	--

Wykład	Metody prowadzenia zajęć:	
	Wykład, Dyskusja, Pokaz	
	Metody (sposoby) weryfikacji:	Udział:
	Test	100%
	Warunki zaliczenia przedmiotu:	
	51% punktów możliwych do uzyskania podczas testu	

Semestr 5

Forma zajęć		
Ćwiczenia laboratoryjne	Metody prowadzenia zajęć:	
	Ćwiczenia laboratoryjne, Projekt, Praca w grupie	
	Metody (sposoby) weryfikacji:	Udział:
	Sprawdzian zdolności metodycznych i umiejętności technicznych	60%
	Sprawozdanie	20%
	Projekt	20%
	Warunki zaliczenia przedmiotu:	
	Zrealizowanie minimum 10 zadań laboratoryjnych, oddanie sprawozdań ze zrealizowanych zadań, opcjonalna realizacja projektu jedno lub dwuosobowego i oddanie sprawozdania.	

Efekt uczenia się dla przedmiotu	Metody (sposoby) weryfikacji			
	Test	Sprawozdanie	Projekt	Sprawdzian zdolności metodycznych i umiejętności technicznych
W1	x			x
W2	x	x	x	
W3	x			
U1		x		
U2		x	x	x
K1			x	

5. Literatura

Literatura podstawowa

1. Robin Nixon, PHP, MySQL i JavaScript, Wprowadzenie. Wydanie IV Helion
2. Laura Lemay, Rafe Colburn, Jennifer Kyrnin HTML. CSS i JavaScript dla każdego. Wydanie VII
3. Charles Wyke-Smith, CSS. Witryny internetowe szyte na miarę. Autorytety informatyki. Wydanie III
4. Stephen Radford, Projektowanie nowoczesnych aplikacji sieciowych z użyciem AngularJS i Bootstrapa

Literatura uzupełniająca

1. Keith Wald, Jason Lengstorf, PHP i jQuery. Techniki zaawansowane.
2. Bogdan Brinzarea-Iamandi, Cristian Darie, Audra Hendrix, AJAX i PHP. Tworzenie interaktywnych aplikacji internetowych.
3. Anna Kempa, Tomasz Staś, Wstęp do programowania w C#.

6. Nakład pracy studenta - bilans godzin i punktów ECTS

Aktywność studenta		Obciążenie studenta Liczba godzin
Zajęcia prowadzone z bezpośrednim udziałem nauczyciela akademickiego lub innych osób prowadzących zajęcia	Wykład	9
	Ćwiczenia laboratoryjne	18
Praca własna studenta	Przygotowanie do zajęć	14
	Konsultacje	2
	Praktyka (praca własna studenta)	40
	Przygotowanie do zaliczenia	15
	Przygotowanie projektu	10
	Przygotowanie sprawozdania	12
Łączny nakład pracy studenta		120
Liczba punktów ECTS		4

* Godzina (dydaktyczna) oznacza 45 minut