Subject card

Subject name and code	Internet Applications Evaluation in Medicine, PG_00068231					
Field of study	Biomedical Engineering					
Date of commencement of studies	October 2025		Academic year of realisation of subject		2027/2028	
Education level	first-cycle studies		Subject group		Optional subject group Subject group related to scientific research in the field of study	
Mode of study	Full-time studies		Mode of delivery		at the university	
Year of study	3		Language of instruction		Polish	
Semester of study	5		ECTS credits		3.0	
Learning profile	general academic profile		Assessment form		exam	
Conducting unit	Department of Biomedical Engineering -> Faculty of Electronics Telecommunications and Informatics -> Faculties of Gdańsk University of Technology					
Name and surname of lecturer (lecturers)	Subject supervisor		dr inż. Anna Węsierska			
	Teachers					
Lesson types	Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
	Number of study hours	30.0	0.0	0.0	15.0	0.0	45
	E-learning hours included: 0.0					
Learning activity and number of study hours	Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
	Number of study hours	45	4.0	26.0	75
Subject objectives	The aim of the course is to equip students with knowledge and skills in the fundamental methods and techniques of web application development, as well as their practical application within a project.					

Learning outcomes	Course outcome	Subject outcome	Method of verification
	[K6_W04] knows and understands, to an advanced extent, the principles, methods and techniques of programming and the principles of computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study, and organisation of systems using computers or such devices	The student has acquired knowledge in the following areas: - installing and configuring a development environment for selected programming languages (e.g., Python Flask, Java, JavaScript, PHP, HTML); - installing and configuring a web server environment along with basic services; - developing programs that run in a web browser environment; - creating applications based on the client–server model, including both client-side and server-side programming; - understanding architectures based on the client–server model, such as serverless architecture and microservices architecture; - programming for the generation of dynamic web content; - synchronizing shared resources in multithreaded environments; - synchronizing threads in multithreaded programming; - integrating web applications with databases; - applying artificial intelligence solutions in web applications (e.g., chatbots, content and service personalization); - using AI tools in the development of web applications.	[SW1] Assessment of factual knowledge
	[K6_U04] can apply knowledge of programming methods and techniques as well as select and apply appropriate programming methods and tools in computer software development or programming devices or controllers using microprocessors or programmable elements or systems specific to the field of study	Learning outcomes include the ability to: - install and configure a development environment for a chosen programming language (e.g. Python/Flask, Java, JavaScript, PHP, HTML), - install and configure a web server environment with basic services, - create a program based on the client-server model (e.g. Python/Flask, Java, JavaScript, PHP, HTML), - develop a program that runs in a web browser environment, - develop a simple program that generates dynamic content for web pages, - develop a simple program that uses parallel processing (e.g. multithreading), - develop a simple program that integrates a web application with a database, - develop a simple program that enables the use of artificial intelligence solutions in web applications.	[SU1] Assessment of task fulfilment

Subject contents	Course content – lecture
	Lecture: 1. Examples of web application use in medicine. 2. Basic requirements for web applications in the medical field. 3. Overview of trends in web application development, especially in the medical domain. 4. Web technologies and environments (e.g. Node.js, Python Flask, Python Django). 5. Examples of cloud service providers and their compatibility with selected web technologies (e.g. Heroku, Google Cloud Platform). 6. Overview of the process of deploying applications for end users. Environment configuration. Starting the application server (e.g. Gunicorn). 7. The client-server model. 8. Client-side web application programming (e.g. JavaScript). 9. Server-side web application programming (e.g. JavaScript). 10. Dynamic web page content. 11. Overview of architectures based on the client-server model, e.g. serverless architecture, microservices architecture. 12. Content Management Systems (CMS) in websites (e.g. WordPress). 13. Database technologies used in web services. 14. Parallel programming in web applications (multiprocessing vs multithreading). 15. Use of multithreading in web applications. 16. Shared resource synchronization in multithreaded environments. 17. Thread synchronization in multithreaded programming. 18. Integration of web applications with databases. 19. Requirements for critical web applications. Examples of real-time web applications and their use in medicine. 20. Video streaming software in websites. 21. Application of AI-based solutions in web applications (e.g. chatbots, content and service personalization). 22. Use of AI tools in the web application development process. **Project:** 1. Defining Requirements for a Sample Web Application in Medicine

	2. Implementation of a Dynamic Website for a Sample Medical Application Using a Selected Technology
	3. Implementation of Dynamic Web Page Content
	4. Integration of Web Applications with Databases
	5. Implementation of Advanced Features in Web Applications (e.g., use of multithreading, real-time data processing/streaming, application of artificial intelligence tools)
	6. Hosting the Web Application
Prerequisites and co-requisites	Methods and techniques of programming 1. The construction program in the programming of structural 1.1. Variables, data types, functions 1.2. Control statements 1.3. Compilation and execution of programs 1.4. Basic data structures 1.5. Ability to move from ideas by the algorithm to 2. Construction of the program in object-oriented programming 2.1. Designing and writing classes 2.2. Creating and using objects 2.3. Elements of object-oriented paradigm (abstraction, encapsulation, inheritance, polymorphism) 2.4. The use of class libraries High-level programming languages

Assessment methods and criteria	Subject passing criteria	Passing threshold	Percentage of the final grade
	Project	51.0%	60.0%
	Lecture - tests/exam	51.0%	40.0%

Recommended reading	Basic literature	Relan, Kunal. "Building REST APIs with Flask." *Building REST APIs with Flask* (2019).
		Rhodes, Brandon, and John Goerzen. *Foundations of Python network programming*. Apress, 2014.
		Reilly, David, and Michael Reilly. *Java network programming and distributed computing*. Addison-Wesley Professional, 2002
		Ballard P., Moncur M., Ajax, JavaScript i PHP. Intensywny trening, Helion 2009 Eckel B., Thinking In Java, edycja polska, Helion 2006 W3C, Rekomendacje XML i HTML, www.w3.org Welling L., Thomson L., PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty, Helion 2005
	Supplementary literature	No requirements
	eResources addresses	

Example issues/ example questions/ tasks being completed	
Practical activites within the subject	Not applicable

Document generated electronically. Does not require a seal or signature.