



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Content Management Systems [S2Bioinf1>SZT]

Course

Field of study
Bioinformatics

Year/Semester
2/3

Area of study (specialization)
–

Profile of study
general academic

Level of study
second-cycle

Course offered in
polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
30

Laboratory classes
30

Other (e.g. online)
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

dr inż. Marcin Borowski
marcin.borowski@put.poznan.pl

Lecturers

Prerequisites

The student starting this course should have basic knowledge of structured and object-oriented programming, programming using the MVC scheme, basic knowledge of internet technologies (HTML, CSS, JS), and basic knowledge of database design. Should have the ability to solve basic problems related to the process of designing IT systems and the ability to obtain information from the indicated sources. Should also understand the need to expand their competences / be ready to cooperate within the team. Moreover, in terms of social competences, the student must present such attitudes as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. Provide students with basic knowledge on the design of content management systems, the flow of information in such applications, technologies used in their construction. 2. Developing students' skills in solving problems related to the design of web and mobile applications, the use of Open Source solutions, frameworks, and libraries supporting the construction of such solutions. 3. Shaping students' teamwork skills and independence in solving problems.

Course-related learning outcomes

Knowledge:

Student:

- has an organized, theoretically founded knowledge of network technologies and internet applications related to the construction of CMS
- has detailed knowledge related to selected issues in the field of IT used in the construction of content management systems (XML, REST, data repositories, processes in CMS)
- has knowledge of development trends in IT and in selected related disciplines - technologies used in CMS construction
- has knowledge about the life cycle of IT systems (content management systems), and the life cycle of data processed in IT

Skills:

Student:

- can, when formulating and solving engineering tasks, integrate knowledge from various areas of computer science (and if necessary knowledge from other scientific disciplines) as well as knowledge from the area of operation of a given CMS and apply a systemic approach, also taking into account non-technical aspects
- can assess the usefulness and the possibility of using new technological achievements (methods, tools, libraries, frameworks, services) and new IT products
- can use analytical, simulation and experimental methods (such as: estimating the number of requests to CMS, load the server with SQL queries) to formulate, solve engineering tasks and simple research problems, can correctly design and implement efficient mechanisms
- can make a critical analysis of the existing technical solutions used by the enterprise and propose their improvements (improvements) in the form of CMS components
- can, in accordance with the given specification, taking into account non-technical aspects and requirements - design a complex CMS and implement this project - at least in part - using appropriate methods, techniques and tools, including adapting existing or developing new tools for this purpose

Social competences:

Student:

- understands that in computer science knowledge and skills very quickly become obsolete, in particular internet and mobile technologies
- understands the need to use the latest technological achievements and knows examples and understands the causes of malfunctioning of CMS systems, which may lead to serious financial, image or social losses

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:

Formative assessment

- a. Lecture: based on activity during the interactive parts of the lectures;
- b. laboratory: based on the assessment of the current progress in the implementation of tasks;

Summative assessment

a. Lecture:

- assessment of the acquired knowledge and skills shown in the exam - a test conducted using an internet application; general and problem-related questions (the student can use any teaching materials; a selection test with about 30 questions; pass according to the following criteria of the obtained result:

> 50% - 3.0

> 60% - 3.5

> 70% - 4.0

> 80% - 4.5

> 90% - 5.0

- discussion of the exam results

b. Laboratory:

- verification of the assumed learning outcomes realized by
- students' assessment and defense of the prepared tasks - 5 small projects;

When assigning the final grade, the student may obtain an increase in grade for:

- discussing additional aspects of the presented issues, not presented during classes;

- using skills and knowledge from outside the study program to solve the tasks performed;
- help in improving teaching materials related to the subject.

Programme content

Lecture:

Basic information on CMS. Types of content management systems. Overview of the components of content management systems, components and processing methods. Basics of XML, DTD, XML Scheme, Relax NG. Structure and syntax of HTML5, CSS3. What's new in JavaScript in the context of HTML5. Business processes, reminder of subjects from previous years. Information on process management systems.

Laboratory:

Laboratory classes are conducted in the form of 2-hour exercises in the laboratory. Classes are carried out independently by students or by 2-person teams of students. The laboratory program covers the following topics: The use of existing content management systems to quickly launch the base (stub) of a web application - based on WebCMS (on the example of CMS Made Simple). Building an application using Django technology. Using the basic components, structures and structures available in the framework. Create your components. Communication with the server via the HTTP, REST and Web Services protocols.

Teaching methods

Lecture: multimedia presentation, illustrated with examples given on the board.

Laboratory exercises: multimedia presentation, presentation illustrated with examples given on the whiteboard, live coding, and carrying out the tasks given by the teacher - practical exercises.

Bibliography

Basic

1. Technical documentation of the mentioned tools available on the Internet
2. Document Engineering, Robert J. Glushko, Tim McGrath, MIT Press, Cambridge, 2005
3. <https://www.w3schools.com/> (wybrane zagadnienia)
4. Dokumentacja Django (<http://djangoproject.com>)
5. Dokumentacja Python (<http://python.org>)

Additional

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50