



COURSE DESCRIPTION CARD - SYLLABUS

Course name

Internet applications [S2ETI1>AI]

Course

Field of study

Education in Technology and Informatics

Year/Semester

1/1

Area of study (specialization)

–

Profile of study

general academic

Level of study

second-cycle

Course offered in

Polish

Form of study

full-time

Requirements

compulsory

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

In terms of knowledge, the student should have basic knowledge of programming structured and object-oriented programming, basic knowledge of web technologies and basic knowledge of designing and using databases. He or she should have the ability to solve basic problems related to the process of designing information systems and the ability to obtain information from indicated sources. In terms of social competences, the student should demonstrate such attitudes as honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. Provide students with basic knowledge on designing web applications, information flow in such applications, technologies used in their construction. 2. Developing students' skills in solving problems related to designing web applications, using 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:

1. has ordered, theoretically founded general knowledge in the field of algorithms, computer systems architecture, operating systems, network technologies, programming languages, graphics, artificial intelligence, databases, decision support, learning systems and software engineering [k2_w10].
2. has theoretically founded detailed knowledge related to selected issues in the field of computer science [k2_w11].
3. knows the basic methods, techniques and tools used in solving complex engineering tasks in the selected area of computer science and technology [k2_w07].

Skills:

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie, w szczególności korzystać z różnego rodzaju dokumentacji technicznych oraz api narzędzi wykorzystywanych w trakcie zajęć praktycznych dotyczących budowy aplikacji internetowych oraz mobilnych [k2_u04].
2. potrafi wykorzystać nabytą wiedzę matematyczną do opisu procesów, tworzenia modeli oraz zapisu algorytmów [k2_u01].
3. potrafi zgodnie z zadaną specyfikacją zaprojektować urządzenie, system informatyczny lub proces używając właściwych metod, technik i narzędzi [k2_u22].

Social competences:

1. understands that knowledge and skills become obsolete very quickly in computer science - understands the need to learn throughout life. he can inspire and organize the learning process of other people [k2_k01].
2. is able to interact and work in a group, assuming various roles in it, in particular during the implementation of it projects [k2_k03].
3. can set priorities for the implementation of tasks defined by himself or others, in particular during the implementation of it projects [k2_k04].

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:

1. In the field of lectures:
 - a. on the basis of answers to questions about the material discussed in previous lectures,
2. in the field of laboratories / exercises:
 - a. on the basis of an assessment of the current progress in the implementation of tasks,

Summative assessment:

- a) in the field of lectures, verification of the assumed learning outcomes is carried out by:
 1. assessment of the knowledge and skills shown in the problematic test (the student can use any teaching materials) / in the form of a selection test (30 questions for a total of 60 points (2 points for the correct answer), passing from 60% + 1)
 - discussion of the results of the test,
 - b) in the field of laboratories / exercises, verification of the assumed learning outcomes is carried out by:
 1. assessment and "defense" by the student of the report on the implementation of the project,
- Obtaining additional points for activity during classes, especially for:
1. discuss additional aspects of the issue,
 - the effectiveness of applying the acquired knowledge while solving a given problem,
 1. remarks related to the improvement of teaching materials, identifying students' perceptual difficulties enabling ongoing improvement of the teaching process.

Programme content

Lecture:

The lecture programme covers the following topics:

Structure and syntax of HTML5, CSS3, JavaScript. Basics of node.js, express.js and some current frontend tools.

Laboratory:

Laboratory classes are conducted in the form of sixteen hour-long classes held in a computer laboratory. The first class is designed to familiarise students with

rules of using the laboratory and passing the exercises.

The syllabus of the laboratory classes includes the following topics:

Creating pages in HTML5. Creating CSS3, LESS, SASS stylesheets. Basics of the language JavaScript and auxiliary libraries. Building applications using the discussed modern front-end technologies.

Course topics

Lecture:

- Structure and syntax of HTML5, CSS3, JavaScript.
- Basics of Node.js and Express.js.
- Current frontend tools.

Lab:

- Creating pages in HTML5.
- Creation of CSS3 stylesheets, LESS, SASS.
- Fundamentals of JavaScript and supporting libraries.
- Building applications with modern frontend technologies.

Teaching methods

Lecture: multimedia presentation, illustrated by examples given on the blackboard.

Laboratory exercises: multimedia presentation, illustrated by examples given on the blackboard, live coding and performance of tasks given by the instructor - practical exercises.

Bibliography

- Bootstrap Framework – <http://getbootstrap.com>
- TailwindCSS - <https://tailwindcss.com/>
- Javascript - <https://www.javascript.com/>
- Svelte - <https://svelte.dev/>

Breakdown of average student's workload

	Hours	ECTS
Total workload	77	4,00
Classes requiring direct contact with the teacher	65	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,00