



## COURSE DESCRIPTION CARD - SYLLABUS

Course name

Internet applications [S2ETI2>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

0

Tutorials

0

Projects/seminars

0

### Number of credit points

4,00

### Coordinators

dr inż. Marcin Borowski

marcin.borowski@put.poznan.pl

### Lecturers

dr inż. Marcin Borowski

marcin.borowski@put.poznan.pl

### Prerequisites

Students should have basic knowledge of structural and object-oriented programming, Internet technologies, database design and use. Problem-solving skills and the ability to use literature and technical documentation are essential. In terms of social skills, reliability, responsibility, creativity and personal culture are expected.

### Course objective

1. Familiarising students with the principles of web application design, data flow and the technologies used.
2. Developing skills in creating solutions based on open source frameworks and libraries.
3. Improving teamwork, independence and project planning skills.

### Course-related learning outcomes

Knowledge:

1. Has knowledge of engineering methods, techniques and tools used in the design and implementation of web applications, including frameworks, libraries and design patterns used in a network environment
2. Knows the principles of designing web application architecture in a systemic approach, including frontend and backend components, and their integration with databases

3. Has structured and in-depth knowledge of web programming languages, the principles of HTTP protocols, client-server technology and web application security
4. Is familiar with current trends and directions in the development of modern internet technologies, including the use of open source frameworks and tools for application automation and testing .

#### Skills:

1. Can analyse user requirements and design complex web applications, selecting appropriate technologies and programming tools (e.g. HTML5, CSS3, JavaScript, Tailwindcss, Sveltekit)
2. Is able to obtain and interpret information from technical documentation, code repositories and industry sources in Polish and English in order to improve the solutions created
3. Is able to apply advanced IT techniques to build and test web applications, including tools for automation, versioning and environment integration (e.g. Git, Vite, NPM/Bun)
4. Is able to plan and carry out the process of testing, implementing and documenting the application being developed, analyse errors and introduce improvements to the source code
5. Is able to communicate effectively within a project team, presenting solutions to technical problems and design justifications in a manner understandable to both technical and non-technical audiences
6. Is able to lead a small project team implementing a web application and distribute tasks according to the competences of team members

#### Social competences:

1. Is ready to independently update their knowledge of new internet technologies and professional development in the field of software engineering
2. Is prepared to work in project teams, open to constructive criticism and ready to share knowledge and experience
3. Demonstrates responsibility for the quality and security of developed applications, complies with the principles of professional ethics and the protection of copyright and intellectual property

### Methods for verifying learning outcomes and assessment criteria

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.
- Assessment criteria/assessment: in accordance with the study regulations

### Programme content

The curriculum covers the basics of web technologies, including HTML5, CSS3, and JavaScript, along with the use of modern libraries and frameworks such as React and SvelteKit. Students learn the principles of creating websites and web applications, as well as the basics of programming in JS/TS.

### Course topics

The lecture program covers the following topics:

Fundamentals of web technologies. Structure and syntax of HTML5 and CSS3. Fundamentals of

JavaScript, use of JS libraries. Building applications using React/SvelteKit.

Laboratory exercises are conducted in the form of fifteen two-hour classes held in a computer lab. The first class is devoted to familiarising students with the rules for using the lab and completing the exercises. The exercises are carried out by teams of two students. The laboratory programme covers the following topics:

Creating pages in HTML5. Creating CSS3 style sheets. Using JavaScript libraries. Programming in JS/TS. Building applications using modern frameworks and libraries.

### Teaching methods

1st lecture: multimedia presentation, presentation illustrated with examples given on the board, solving problems, multimedia show, demonstration.

2. laboratory classes: practical exercises, discussion, team work, multimedia show, case studies, demonstration.

### Bibliography

Basic:

1. HTML5 - <https://developer.mozilla.org/en-US/docs/Web/HTML>
2. CSS3 - <https://developer.mozilla.org/en-US/docs/Web/CSS>
3. Tailwindcss Framework - <https://tailwindcss.com/>
4. SvelteKit Documentation - <http://svelte.dev>

Additional:

-

### Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	62	2,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	38	2,00