COURSE OUTLINE

(1) GENERAL

SCHOOL	Engineering		
ACADEMIC UNIT	Department of Computer Engineering and Informatics		
LEVEL OF STUDIES	Undergraduate		
COURSE CODE	**26EX581**	SEMESTER	**Winter**
COURSE TITLE	Advanced Web Technologies and Architectures		

INDEPENDENT TEACHING ACTIVITIES *if credits are awarded for separate components of the course, e.g. lectures, laboratory exercises, etc. If the credits are awarded for the whole of the course, give the weekly teaching hours and the total credits*	WEEKLY TEACHING HOURS	CREDITS
Lectures, Tutorials, Laboratory	2 (Lec), 1 (Tut), 2 (Lab)	5
Add rows if necessary. The organisation of teaching and the teaching methods used are described in detail at (d).		

COURSE TYPE *general background, special background, specialised general knowledge, skills development*	Specialised general knowledge, Skills development
PREREQUISITE COURSES:	Recommended prerequisite knowledge: Good familiarity with the courses • CEID_22Y103 Introduction to Programming • CEID_23Y106 Object-Oriented Programming • CEID_23Y205 Introduction to Algorithms • CEID_23Y210 Data Structures • CEID_23Y206 Principles of Programming Languages and Compilers • CEID_24Y334 Databases • CEID_24Y387 Computer Networks • CEID_24Y338 Web Programming and Systems or equivalents.
LANGUAGE OF INSTRUCTION and EXAMINATIONS:	Greek. The course may be offered in English for Erasmus students.
IS THE COURSE OFFERED TO ERASMUS STUDENTS	Yes
COURSE WEBSITE (URL)	[pending]

(2) LEARNING OUTCOMES

Learning outcomes

The course learning outcomes, specific knowledge, skills and competences of an appropriate level, which the students will acquire with the successful completion of the course are described.

Consult Appendix A

- *Description of the level of learning outcomes for each qualifications cycle, according to the Qualifications Framework of the European Higher Education Area*
- *Descriptors for Levels 6, 7 & 8 of the European Qualifications Framework for Lifelong Learning and Appendix B*
- *Guidelines for writing Learning Outcomes*

Upon successful completion of the course, students will be able to achieve the following Learning Outcomes:

1. Knowledge (Assimilation of information, theories, and principles)
    a. Describe the fundamental operating principles of the World Wide Web, including protocols (HTTP/HTTPS), caching mechanisms, and content redundancy systems.
    b. Recognize the characteristics of modern development architectures (monolithic, microservices, cloud computing, serverless), as well as security (OWASP), usability, and accessibility (WCAG) standards.

2. Skills (Application of knowledge, practical ability, and mental process)
    a. Develop modern, interactive, and accessible Progressive Web Applications (PWAs) and real-time applications using popular front-end (React, Vue, Angular) and back-end (Node.JS, Python) frameworks.
    b. Apply performance optimization techniques (lazy loading, code splitting, efficient queries) and manage data using non-relational databases.
    c. Utilize cache consistency algorithms and content validation protocols, while integrating Secure Software Development Life Cycle (SSDLC) practices.

3. Competences (Problem-solving, knowledge transfer, autonomy, and decision-making)
    a. Design comprehensive and scalable large-scale systems, autonomously selecting the appropriate software technologies to solve complex architectural problems of World Wide Web applications.
    b. Critically evaluate the performance, security, and usability of complex systems based on best practices derived from case studies of modern services, making informed decisions on their optimization strategy.
    c. Creatively integrate emerging technologies (such as Artificial Intelligence, Machine Learning in JavaScript, and WebGL graphics) to provide innovative solutions, responsibly adapting to the ever-changing demands of the information society.

1. General Competences

Taking into consideration the general competences that the degree-holder must acquire (as these appear in the Diploma Supplement and appear below), at which of the following does the course aim?

Search for, analysis and synthesis of data and information, with the use of the necessary technology
Adapting to new situations
Decision-making
Working independently
Team work
Working in an international environment
Working in an interdisciplinary environment
Production of new research ideas
Project planning and management
Respect for difference and multiculturalism
Respect for the natural environment
Showing social, professional and ethical responsibility and sensitivity to gender issues
Criticism and self-criticism
Production of free, creative and inductive thinking
……
Others…
…….

- Search for, analysis and synthesis of data and information, with the use of the necessary technology
- Adapting to new situations
- Decision-making
- Team work
- Project planning and management
- Production of free, creative and inductive thinking

(3) SYLLABUS

The course is divided into two main parts: Advanced Web Technologies and Web Architectures.

Advanced Web Technologies

1. Asynchronous JavaScript, APIs, and Promises
2. Introduction to TypeScript and modern JS application development frameworks (React, Vue, Angular)

3. Modern back-end development frameworks (Node.JS, Express.js, Python/Django)
4. Full-stack development (Next.js)
5. Front-end Optimization Techniques (Rendering optimization, code splitting, lazy loading, image optimization, browser caching/storage, performance profiling tools)
6. Use of NoSQL databases for Web applications (MongoDB, Firebase)
7. Back-end Optimization Techniques (Efficient database queries, caching strategies, optimizing API responses, load testing basics)
8. Real-time applications (WebSockets, Event-driven architectures)
9. Single Page Applications & Multi Page Applications
10. Design and development of Progressive Web Applications (PWAs)
11. Web Usability (human-centered design, usability heuristics, usability evaluation methods)
12. Web Accessibility (WCAG guidelines, ARIA attributes, accessibility-driven development techniques, evaluation tools)
13. Graphics Programming with JavaScript (WebGL for 2D/3D graphics, AR/VR applications)
14. Emerging Web technologies (Artificial Intelligence & Machine Learning with JavaScript technologies)

Web Architectures

1. Introduction to the basic concepts of caches and content redundancy.
2. HTTP protocol operations (simple, persistent & pipelining), headers for supporting cache and content redundancy operations, and data transfer security with encryption (SSL/TLS, HTTPS).
3. Cache performance impacts and metrics on web operation, TCP connection splitting, and storing TCP connections in caches.
4. Cache consistency management algorithms and techniques, invalidation protocols (leases, subscriptions, leased subscriptions) and cache content validation (tugs-of-war / polling-vs-push trade-offs, adaptive lifetime/TTL adjustment, piggyback cache validation).
5. Redundancy systems (static/transparent, partial/full mirroring and combinations, request distribution techniques with or without content awareness).
6. Deployment Architectures (monolithic, microservices, serverless computing)
7. Scalable Web Applications (Cloud computing, containerization, Kubernetes)
8. Advanced Web Security Topics (OWASP top 10, SSDLC, advanced user authentication methods, public key infrastructures)
9. Case studies of large-scale Web systems (e.g., Netflix, Spotify, Instagram, etc.)

(4) TEACHING and LEARNING METHODS - EVALUATION

DELIVERY *Face-to-face, Distance learning, etc.*	Ex cathedra.
USE OF INFORMATION AND COMMUNICATIONS TECHNOLOGY *Use of ICT in teaching, laboratory education, communication with students*	The slides of the course and additional supplementary material are freely available from the course's website. Communication with students is done through a dedicated e-forum.
TEACHING METHODS *The manner and methods of teaching are described in detail.* *Lectures, seminars, laboratory practice, fieldwork, study and analysis of bibliography, tutorials, placements, clinical practice, art workshop, interactive teaching, educational visits, project, essay writing, artistic creativity, etc.* *The student's study hours for each learning activity are given as well as the hours of non-directed study according to the principles of the ECTS*	

Activity	*Semester workload*
Lectures	2x13 = 26
Tutorials (exercises)	1x13 = 13
Laboratory exercises	2x13 = 26
Individual study, preparation and problem solving	2x13 = 26
Weekend study	2x13 = 26
Study during the 3 “empty weeks” (2 weeks of	8x3 = 24

	vacation and 1 week of exam preparation)	
	Course total	141
STUDENT PERFORMANCE EVALUATION *Description of the evaluation procedure* *Language of evaluation, methods of evaluation, summative or conclusive, multiple choice questionnaires, short-answer questions, open-ended questions, problem solving, written work, essay/report, oral examination, public presentation, laboratory work, clinical examination of patient, art interpretation, other* *Specifically-defined evaluation criteria are given, and if and where they are accessible to students.*	• The language of instruction and examination is Greek. Special provisions (lecture notes and examinations in English) can be made for foreign students. • The final grade is based on performance in the final examination (30% of the total grade) and course project (70% of the total grade). The evaluation criteria are posted on the course website. • The final examination is written, of graded difficulty, and includes multiple choice questionnaires and short-answer questions (30%). • During the course, students are assigned a web development project, requiring the application of multiple web programming skills (70%).	

(5) ATTACHED BIBLIOGRAPHY

- Suggested bibliography in English:

- Ricardo Puttini, Thomas Erl, Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture 2nd edition, Pearson, 2023
- Robin Nixon, Learning PHP, MySQL & JavaScript - A Step-by-Step Guide to Creating Dynamic Websites, 6th Edition, O'Reilly, 2023

- Freely accessible web resources and content, e.g.
  - https://www.w3schools.com/
  - https://developer.mozilla.org/en-US/docs/Learn_web_development

- Related academic journals:

ACM Transactions on the Web (TWEB)
World Wide Web: Internet and Web Information Systems – Springer
ACM Transactions on Internet Technology (TOIT)
IEEE Internet Computing